



TUGAS AKHIR - EE 184801

**RANCANG BANGUN SISTEM PEMBAYARAN TUNAI
PADA MESIN PENJUAL SWALAYAN (VENDING
MACHINE)**

Fidelis Prasetyo
NRP 07111440000202

Dosen Pembimbing
Dr. Ir. Hendra Kusuma, M.Eng.Sc.
Ir. Tasripan, MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



TUGAS AKHIR - EE 184801

**RANCANG BANGUN SISTEM PEMBAYARAN TUNAI
PADA MESIN PENJUAL SWALAYAN (VENDING
MACHINE)**

Fidelis Prasetyo
NRP 07111440000202

Dosen Pembimbing
Dr. Ir. Hendra Kusuma, M.Eng.Sc.
Ir. Tasripan, MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



FINAL PROJECT - EE 184801

**DESIGN AND IMPLEMENTATION OF CASH PAYMENT
SYSTEM ON SELF-SERVICE VENDING MACHINE**

Fidelis Prasetyo
NRP 07111440000202

Supervisor
Dr. Ir. Hendra Kusuma, M.Eng.Sc.
Ir. Tasripan, MT.

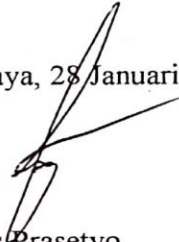
ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2019

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Rancang Bangun Sistem Pembayaran Tunai pada Mesin Penjual Swalayan (Vending Machine)” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 28 Januari 2019



Fidelis Prasetyo
NRP. 07111440000202

**RANCANG BANGUN SISTEM PEMBAYARAN TUNAI PADA
MESIN PENJUAL SWALAYAN (VENDING MACHINE)**

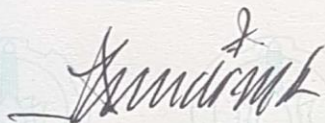
TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik
Pada**

**Bidang Studi Elektronika
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

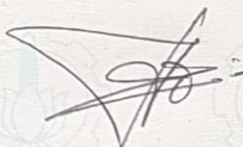
Menyetujui

Dosen Pembimbing I



**Dr. Ir. Hendra Kusuma, M.Eng.Sc.
NIP: 196409021989031003**

Dosen Pembimbing II



**Ir. Tasripan, MT.
NIP: 196204181990031004**



.....*Halaman ini sengaja dikosongkan*.....

RANCANG BANGUN SISTEM PEMBAYARAN TUNAI PADA MESIN PENJUAL SWALAYAN (VENDING MACHINE)

Nama : Fidelis Prasetyo
Pembimbing 1 : Dr. Ir. Hendra Kusuma, M.Eng.Sc.
Pembimbing 2 : Ir. Tasripan, MT.

ABSTRAK

Untuk meningkatkan omset penjualan suatu barang tanpa membutuhkan tenaga penjual lebih, dibutuhkan suatu mesin penjual swalayan yang mampu menjual barang secara otomatis. Mesin penjual swalayan terdiri dari sistem pembayaran tunai dan pengeluaran barang yang dijual. Pada tugas akhir ini, telah dikembangkan sistem pembayaran tunai pada mesin penjual swalayan yang mampu menarik dan mendorong uang, merekognisi nominal uang, dan memberikan uang kembalian. Sistem pembayaran tunai yang dibangun pada tugas akhir ini mampu mengenali keautentikan uang kertas rupiah tahun emisi 2016 dengan nominal Rp 10.000,00, Rp 20.000,00, Rp 50.000,00, dan Rp 100.000,00. Metode pengenalan nominal uang berdasarkan citra yang digunakan dalam tugas akhir ini adalah metode *Principal Component Analysis* (PCA) dan *Linear Discriminant Analysis* (LDA). Pengujian *scanner* uang sistem menghasilkan tingkat *error* 0% pada penarikan uang dan 23,3% pada pendorongan uang. Pengujian kembalian uang yang diimplementasikan menghasilkan tingkat *error* 20% untuk Rp 10.000,00 dan 15% pada Rp 50.000,00. Pengujian kedua metode pengenalan uang yang digunakan pada tugas akhir ini menghasilkan tingkat akurasi 97,5% untuk metode PCA dan tingkat akurasi 100% untuk metode LDA. Metode LDA kemudian diimplementasikan ke dalam sistem untuk diuji kembali dan diperoleh akurasi hasil pengujian rekognisi nominal citra uang pada sistem adalah 98,3% dari 60 sampel pengujian. Pengujian akhir meliputi pengujian sistem secara keseluruhan dari penarikan uang sampai pengembalian uang. Dari pengujian dapat disimpulkan bahwa sistem dapat bekerja sebagai sistem pembayaran tunai pada mesin penjual swalayan dengan baik.

Kata kunci: LDA, Mesin Penjual Swalayan, PCA, Rekognisi Nominal Citra Uang.

.....*Halaman ini sengaja dikosongkan*.....

DEISGN AND IMPLEMENTATION OF CASH PAYMENT SYSTEM ON SELF-SERVICE VENDING MACHINE

Name : Fidelis Prasetyo
1st Advisor : Dr. Ir. Hendra Kusuma, M.Eng.Sc.
2nd Advisor : Ir. Tasripan, MT.

ABSTRACT

To increase the level of sales of an item without requiring more salespeople, a self-service vending machine is needed which able to sell goods automatically. Self-service vending machines consist of a cash payment system and the expenditure of goods sold. In this final project, a cash payment system that has been implemented in the self-service vending machine will be able to receive and return money, recognize money, and provide change. The cash payment system built in this final project is able to recognize the authenticity of 2016 emission rupiah banknotes with nominal values of Rp 10,000.00, Rp. 20,000.00, Rp. 50,000.00 and Rp. 100,000.00. The method of image-based cash recognition used in this final project is the Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) method. Testing of system's money scanner generates a 0% error rate on money withdrawals and 23.3% on money pushing. Testing of the implemented refunds system produces an error rate of 20% for IDR 10,000.00 and 15% at IDR 50,000.00. The testing of the two money recognition methods used in this final project resulted in an accuracy rate of 97.5% for the PCA method and a 100% accuracy rate for the LDA method. The LDA method was then implemented into the system to be re-tested and obtained the accuracy of the nominal image image recognition test on the system was 98.3% based on 60 test samples. The final test includes testing the whole system from withdrawing money to returning money. From the tests, it can be concluded that the system can work well as a cash payment system on a self-service vending machine.

Keywords: Image-based Currency Recognition, LDA, PCA, Vending Machine.

.....*Halaman ini sengaja dikosongkan*.....

KATA PENGANTAR

Puji syukur atas rahmat yang telah diberikan oleh Tuhan Yang Maha Esa. Karena berkat rahmat, berkat, dan bimbingan-Nya penulis dapat menyelesaikan tugas akhir dengan judul **“RANCANG BANGUN SISTEM PEMBAYARAN TUNAI PADA MESIN PENJUAL SWALAYAN (VENDING MACHINE)”**. Dalam menjalankan proses penelitian ini, penulis mendapatkan bantuan dari berbagai pihak baik berupa dukungan secara moril maupun bantuan secara materiil. Oleh karena itu penulis ingin mengucapkan Terima kasih yang sebesar-besarnya kepada berbagai pihak yang mendukung dan membantu dalam penelitian ini, diantaranya:

1. Dr. Ir. Hendra Kusuma, M.Eng.Sc. dan Ir. Tasripin, MT., selaku dosen pembimbing, atas bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan tugas akhir ini.
2. Dr. Muhammad Rivai, ST., MT., Dr. Ir. Djoko Purwanto, M.Eng., Ronny Mardiyanto, ST., MT., Ph.D., dan Ir. Haris Pringadi, MT. selaku dosen penguji yang telah memberikan kritik dan saran.
3. Orang tua dan keluarga dari penulis yang senantiasa memberikan dukungan baik moril dan materiil.
4. Seluruh dosen dan karyawan Departemen Teknik Elektro ITS yang telah memberikan banyak ilmu dan menciptakan suasana belajar yang nyaman.
5. Teman-teman laboratorium Elektronika Dasar B202 yang senantiasa membantu dan memberikan dukungan dalam mengerjakan penelitian ini.

Penulis sadar bahwa penelitian ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan lebih lanjut.

Surabaya, 28 Januari 2019

Fidelis Prasetyo
NRP. 07111440000202

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR ISI

PERNYATAAN KEASLIAN	i
ABSTRAK.....	v
<i>ABSTRACT</i>.....	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xv
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	2
1.3. Tujuan Penelitian	2
1.4. Batasan Masalah	3
1.5. Metodologi Penelitian	3
1.6. Sistematika Penulisan	5
1.7. Relevansi	6
TINJAUAN PUSTAKA	7
2.1. Mesin Penjual Swalayan (Vending Machine)	7
2.2. Uang Kertas Rupiah Tahun Emisi 2016.....	8
2.3. Arduino Mega 2560	9
2.4. Sinar Ultraviolet (UV)	10
2.5. Motor DC	11
2.6. Modul L298 DC Motor Driver	12
2.7. Motor Stepper.....	13
2.8. Modul ULN2003 Stepper Motor Driver	13
2.9. Modul Sensor Infrared TCRT5000.....	14
2.10. Modul 2-Channel Relay	15
2.11. LCD Display 16x2	15
2.12. Kamera USB	16
2.13. 4x4 Matrix Membrane Keypad	16
2.14. Computer Vision.....	17
2.15. Rekognisi Pola Citra	17
2.16. Principal Component Analysis (PCA)	18
2.17. Linear Discriminant Analysis (LDA)	19
2.18. MATLAB	21
2.19. Tinjauan Pustaka.....	22
PERANCANGAN SISTEM.....	25

3.1.	Diagram Blok Sistem	25
3.2.	Perancangan Perangkat Keras	27
3.2.1.	Scanner Uang	27
3.2.2.	Pengembalian Uang	31
3.2.3.	User Interface	34
3.2.4.	Rangkaian Suplai Tegangan DC	37
3.2.5.	Arduino Mega 2560	37
3.2.6.	PC (<i>Personal Computer</i>)	39
3.2.7.	Bodi Luar Sistem	40
3.3.	Perancangan Perangkat Lunak	40
3.2.1.	Akuisisi Citra Uang	41
3.2.2.	Pemrosesan Citra Digital Uang	42
3.2.3.	Analisa Citra Digital Uang	43
3.2.4.	Sistem Pembayaran Mesin Penjual Swalayan	54
PENGUJIAN DAN ANALISIS		61
4.1.	Realisasi Sistem	61
4.2.	Inisialisasi Sensor IR TCRT5000	62
4.3.	Pengujian Scanner Uang	64
4.4.	Pengujian Pengembalian Uang	64
4.5.	Pengujian Akuisisi Citra Uang	65
4.6.	Pengujian Pemrosesan Citra Digital Uang	66
4.7.	Pengujian Pengenalan Citra Uang	67
4.8.	Pengujian Dengan Uang Palsu	75
4.9.	Pengujian Keseluruhan Sistem	76
PENUTUP		85
5.1.	Kesimpulan	85
5.2.	Saran	85
DAFTAR PUSTAKA		87
LAMPIRAN A		89
LAMPIRAN B		107
LAMPIRAN C		117
BIODATA PENULIS		121

DAFTAR GAMBAR

Gambar 2.1 Contoh vending machine[2]	7
Gambar 2.2 Pecahan uang kertas rupiah tahun emisi 2016[3]	8
Gambar 2.3 Ciri-ciri dan unsur pengaman Rp 100.000,00[3]	9
Gambar 2.4 Arduino Mega 2560	10
Gambar 2.5 Lampu ultraviolet.....	11
Gambar 2.6 Motor DC	12
Gambar 2.7 Modul L298.....	12
Gambar 2.8 Motor stepper.....	13
Gambar 2.9 Modul ULN2003	14
Gambar 2.10 Modul TCRT5000.....	14
Gambar 2.11 Modul 2-channel Relay	15
Gambar 2.12 LCD display 16x2	16
Gambar 2.13 Kamera USB.....	16
Gambar 2.14 4x4 matrix membrane keypad	17
Gambar 2.15 Ilustrasi PCA[25]	19
Gambar 2.16 Ilustrasi LDA[25].....	21
Gambar 2.17 Logo MATLAB[27].....	21
Gambar 3.1 Diagram blok sistem.	25
Gambar 3.2 Rancangan mekanik scanner	28
Gambar 3.3 Antarmuka driver motor L298 dengan Arduino Mega	29
Gambar 3.4 Antarmuka driver motor L298 dengan motor DC	29
Gambar 3.5 Antarmuka TCRT5000 dengan Arduino Mega.....	30
Gambar 3.6 Tampak depan pemasangan kamera dan lampu UV	31
Gambar 3.7 Rancangan pengembalian uang sederhana	32
Gambar 3.8 Antarmuka driver ULN2003 dengan Arduino Mega.....	32
Gambar 3.9 Rangkaian antara motor stepper dan ULN2003	33
Gambar 3.10 Antarmuka 2-channel relay dengan Arduino Mega	33
Gambar 3.11 Antarmuka motor DC, relay, dan rangkaian suplai	34
Gambar 3.12 Rancangan user interface sistem.....	34
Gambar 3.13 Rangkaian antara LCD dengan modul I2C	35
Gambar 3.14 Antarmuka LCD dengan Arduino Mega	35
Gambar 3.15 Antarmuka keypad dengan Arduino Mega	36
Gambar 3.16 Skematik rangkaian LED	36
Gambar 3.17 Skematik rangkaian suplai tegangan DC.	37
Gambar 3.18 Pinout Arduino Mega 2560[28].....	38
Gambar 3.19 Rancangan bodi luar sistem.....	40

Gambar 3.20 Diagram blok klasifikasi citra uang	41
Gambar 3.21 Diagram blok penyusunan matriks database citra latihan dalam subspace PCA.....	44
Gambar 3.22 Penyusunan matriks citra uang gabungan.....	45
Gambar 3.23 Diagram blok penyusunan matriks pengujian dalam subspace PCA	48
Gambar 3.24 Diagram blok penyusunan matriks database citra latihan dalam subspace LDA	49
Gambar 3.25 Diagram blok penyusunan matriks pengujian dalam subspace LDA	52
Gambar 3.26 Flowchart dari scanner uang pada sistem	56
Gambar 3.27 Flowchart pengenalan citra uang pada MATLAB	57
Gambar 3.28 Flowchart program pengembalian uang	59
Gambar 3.29 Flowchart sistem transaksi mesin penjual swalayan	60
Gambar 4.1 (a) Realisasi mekanik dan elektronik sistem (b) Realisasi sistem secara keseluruhan	61
Gambar 4.2 Letak sensor IR depan	62
Gambar 4.3 Letak sensor IR belakang	63
Gambar 4.4 (a) Citra akuisisi tanpa pengaturan. (b) Citra akuisisi dengan pengaturan	66
Gambar 4.5 Mean dari citra latihan dengan metode PCA dan LDA.....	69
Gambar 4.6 (a) Citra awal. (b) Citra setelah dikurangi mean.....	70
Gambar 4.7 Eigenvectors dari database metode PCA.....	70
Gambar 4.8 Eigenvectors dari database metode LDA	71
Gambar 4.9 Perbandingan jarak Euclidean hasil uji PCA dan LDA.....	74
Gambar 4.10 Perbandingan akurasi hasil uji PCA dan LDA.....	74
Gambar 4.11 Tampilan mode rekognisi pada LCD	78
Gambar 4.12 (a) Pengujian dengan uang Rp 100.000,00. (b) Hasil rekognisi pada LCD	78
Gambar 4.13 Tampilan awal mesin penjual swalayan	79
Gambar 4.14 Tampilan LCD ketika keypad ditekan sebelum ada uang yang masuk.....	79
Gambar 4.15 (a) Proses pemberian uang Rp 100.000,00. (b) Posisi uang di dalam scanner uang.....	80
Gambar 4.16 Citra uang yang diakuisisi oleh kamera.....	80
Gambar 4.17 Saldo hasil deteksi pengujian.....	81

DAFTAR TABEL

Tabel 3.1 Pembagian pin Arduino Mega 2560 pada sistem	39
Tabel 3.2 Spesifikasi PC yang digunakan pada sistem	39
Tabel 3.3 Rekognisi berdasarkan posisi baris jarak terkecil	53
Tabel 3.4 Perintah masukan pada <i>keypad</i>	58
Tabel 4.1. Spesifikasi sistem	62
Tabel 4.2. Hasil inisialisasi sensor IR depan dan belakang.....	63
Tabel 4.3. Hasil pengujian <i>scanner</i> uang	64
Tabel 4.4. Hasil pengujian pengembalian uang	64
Tabel 4.5. Pengaturan kamera untuk akusisi citra uang	65
Tabel 4.6. Hasil pengujian pemrosesan citra digital uang.....	67
Tabel 4.7. Deskripsi citra latihan	68
Tabel 4.8. Pengujian citra uang dengan <i>database</i> PCA	72
Tabel 4.9. Kesimpulan pengujian citra uang dengan <i>database</i> PCA.....	72
Tabel 4.10. Pengujian citra uang dengan <i>database</i> LDA.....	73
Tabel 4.11. Kesimpulan pengujian citra uang dengan <i>database</i> LDA...	73
Tabel 4.12. Pengujian LDA dengan citra uang palsu.....	75
Tabel 4.13. Hasil pengujian rekognisi citra uang	77
Tabel 4.14. Akurasi dengan pengujian ulang.....	77

.....*Halaman ini sengaja dikosongkan*.....

BAB I

PENDAHULUAN

Tugas akhir merupakan suatu penelitian yang dilakukan sebagai persyaratan akademik untuk mendapatkan gelar sarjana teknik di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Tugas akhir ini mengambil topik perancangan dan pembangunan sistem pembayaran uang kertas untuk mesin penjual swalayan atau *vending machine*.

Pada bab ini membahas mengenai hal-hal yang mendahului pelaksanaan tugas akhir. Hal tersebut meliputi latar belakang, perumusan masalah, tujuan penulisan, batasan masalah, metodologi penelitian, sistematika penulisan, dan relevansi.

1.1. Latar Belakang

Seiring dengan meningkatnya tingkat kesibukan penduduk, permintaan akan kegiatan jual beli juga terus meningkat, terutama di lokasi-lokasi yang padat dengan manusia seperti bandara, stasiun, terminal, halte, maupun tempat-tempat umum lainnya. Tetapi di sisi lain, tidak semua penjual produk konvensional mampu memberikan pelayanan yang tersedia sepanjang hari karena terbatasnya sumber tenaga manusia yang berperan sebagai penjual. Padahal hal ini merupakan peluang bagi pihak penjual barang untuk menambah omset penjualan. Di sisi lain, waktu merupakan suatu hal berharga bagi setiap individu, terlebih lagi di tengah kesibukan sehari-hari. Maka dari itu, kegiatan jual beli diusahakan dapat berlangsung dalam durasi sesingkat mungkin.

Untuk menjawab tantangan tersebut, digunakan suatu mesin penjual swalayan atau *vending machine*, yaitu sebuah mesin otomatis yang dapat menggantikan fungsi penjual manusia dalam kegiatan jual beli. Dengan adanya *vending machine*, kegiatan jual beli dapat berlangsung tanpa membutuhkan kehadiran tenaga penjual manusia, dapat dilakukan kapan saja selama stok barang tersedia, dan dapat selesai dalam waktu yang singkat. Selain hal-hal di atas, *vending machine* juga dapat menghindari kecurangan dalam transaksi seperti penggunaan uang palsu karena *vending machine* dirancang agar dapat mendeteksi keaslian uang.

Dalam penelitian ini akan dibangun suatu sistem pembayaran tunai pada mesin penjual swalayan atau *vending machine* yang mampu menerima uang kertas rupiah, mengenali nominal uang kertas dan memastikan keautentikan uang kertas, menghitung jumlah transaksi yang

dilakukan, dan memberikan uang kembalian dalam bentuk uang kertas. Sistem *vending machine* terdiri dari mekanisme penerimaan uang kertas, mekanisme pengembalian uang kertas, serta sistem rekognisi nominal uang kertas. Pada tugas akhir ini, rekognisi nominal uang kertas dilakukan berdasarkan analisa citra uang kertas dengan 2 metode analisa, yaitu *Principal Component Analysis* (PCA) dan *Linear Discriminant Analysis* (LDA). PCA dan LDA adalah suatu metode yang sering digunakan dalam sistem rekognisi biometri wajah berbasis citra. Pada dasarnya, PCA adalah metode analisa yang melakukan reduksi dimensi dari kelompok data (*dataset*) masukan dan dalam waktu yang sama menjaga karakteristik *dataset* masukan[1]. Sedangkan metode analisa LDA adalah metode transformasi matriks yang melakukan transformasi untuk memaksimalkan separasi antar kelas data. Perangkat keras sistem terdiri dari mekanik sistem, motor DC, motor *stepper*, kamera USB eksternal, lampu ultraviolet, *Personal Computer* (PC) dan mikrokontroler Arduino Mega. Perangkat lunak terdiri dari program analisa data *training* dan *testing* citra uang kertas menggunakan MATLAB dan program sistem aktuator pembayaran *vending machine* menggunakan Arduino IDE. Objek penelitian yang digunakan dalam rekognisi uang adalah uang kertas rupiah tahun emisi 2016 dengan nominal pecahan Rp. 100.000,00, Rp. 50.000,00, Rp. 20.000,00, Rp 10.000,00, Rp 5.000,00. Uang kertas kembalian yang disediakan dalam sistem ini adalah pecahan Rp 10.000,00 dan pecahan Rp 5.000,00.

1.2. Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah:

1. Bagaimana mekanisme sistem untuk menerima dan mengembalikan uang kertas.
2. Bagaimana sistem mampu mengambil citra uang kertas dengan tanda air atau *watermark*.
3. Bagaimana sistem mampu melakukan rekognisi nominal dan keautentikan uang kertas.
4. Bagaimana sistem mampu melakukan transaksi dan memberikan kembalian yang sesuai.

1.3. Tujuan Penelitian

Penelitian pada penelitian ini bertujuan sebagai berikut:

1. Merancang mekanik sistem dengan sensor inframerah dan aktuator motor DC untuk menerima dan mengembalikan uang kertas.
2. Merancang mekanik sistem dengan lampu ultraviolet untuk menampilkan tanda air (*watermark*) uang kertas dan kamera USB untuk mengambil citra uang, serta program untuk mengakuisisi citra dengan ukuran dan posisi yang sesuai.
3. Melakukan ekstraksi data citra uang dan rekognisi nominal citra uang.
4. Merancang sistem untuk mengalkulasi jumlah uang yang dikenali dan memberikan kembalian uang kertas sesuai transaksi yang dilakukan.

1.4. Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Uang kertas yang digunakan dalam kondisi layak digunakan.
2. Uang kertas yang digunakan adalah uang kertas rupiah tahun emisi 2016 dengan nominal Rp 100.000,00, Rp 50.000,00, Rp 20.000, dan Rp 10.000,00.
3. Uang kertas kembalian yang tersedia adalah pecahan Rp 10.000,00 dan Rp 50.000,00.
4. Basis data citra latihan yang digunakan berjumlah 112 citra yang terdiri dari 7 sampel citra setiap orientasi untuk masing-masing nominal.

1.5. Metodologi Penelitian

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

Studi literatur merupakan tahap pengumpulan informasi dan referensi dari berbagai sumber sebagai acuan dalam menjalankan proses Tugas Akhir. Referensi dan informasi yang dimaksud diperoleh dari buku, jurnal, tesis, paper, artikel, internet, dan hasil konsultasi dengan dosen pembimbing.

2. Perancangan Mekanik

Pada tahap ini dilakukan perancangan mekanik dari sistem

pembayaran mesin penjual swalayan atau *vending machine*. Tahap yang dilakukan meliputi:

1. Desain mekanik pengambilan uang kertas.
2. Desain mekanik penempatan lampu ultraviolet, kamera, motor, dan sensor.
3. Desain mekanik pengembalian uang kertas.
4. Desain rangka untuk menyatukan sistem.

3. Perancangan Perangkat Lunak

Pada tahap ini dilakukan perancangan dan pembuatan perangkat lunak sistem pembayaran mesin penjual swalayan. Tahap ini meliputi:

1. Merancang program MATLAB untuk akuisisi citra uang dengan posisi yang sesuai.
2. Merancang program MATLAB untuk pemrosesan citra uang yang meliputi *cropping*, *resizing*, dan konversi ke *grayscale*.
3. Merancang program MATLAB untuk melakukan *training* citra uang, melakukan *testing* citra uang, dan mengalkulasi *distance* antara citra *training* dan *testing*.
4. Merancang program Arduino yang mengendalikan seluruh sensor dan aktuator sistem.

4. Pengujian Mekanik Sistem dan Program Rekognisi Uang

Tahap ini dilakukan untuk menguji masing-masing bagian sistem, yaitu mekanik dan program pengenalan citra uang. Aktuator sistem diharapkan dapat menjalankan mekanik sebelum diintegrasikan dengan perangkat lunak. Pada tahap ini juga dilakukan pengujian program rekognisi citra uang agar diperoleh program rekognisi uang dengan tingkat akurasi setinggi mungkin.

5. Implementasi Sistem pada Mikrokontroler

Setelah dilakukan berbagai perancangan di atas, dilakukan implementasi sistem dengan mengintegrasikan seluruh komponen sistem pada mikrokontroler Arduino Mega 2560.

6. Tahap Pengujian Sistem

Pada tahap ini dilakukan pengujian akhir sistem secara keseluruhan. Pengujian sistem meliputi pengujian pengambilan uang, pengujian pengembalian uang, pengujian akuisisi citra uang, pengujian pemrosesan citra uang, pengujian sistem rekognisi

nominal berdasarkan citra uang, dan pengujian sistem secara keseluruhan. Pengujian dilakukan beberapa kali untuk memperoleh data yang dapat dianalisis.

7. Analisa dan Evaluasi

Tahap ini dimaksudkan untuk menganalisis hasil data pengujian dan mengevaluasi kelebihan dan kekurangan sistem. Pada tahap ini juga diharapkan dapat mengidentifikasi masalah-masalah yang ditemui selama proses jalannya penelitian serta mengusulkan solusi terhadap masalah tersebut.

8. Penyusunan Laporan Tugas Akhir

Setelah dilakukan pengujian dan pengambilan data, dilanjutkan dengan penulisan laporan penelitian berdasarkan hasil pengujian sistem untuk merangkum seluruh rangkaian kegiatan penelitian. Laporan tugas akhir merangkum seluruh hal yang berkaitan dengan tugas akhir yaitu meliputi pendahuluan, tinjauan pustaka, perancangan dan pembuatan sistem, pengujian dan Analisa data, serta penutup.

1.6. Sistematika Penulisan

Laporan tugas akhir ini terdiri dari lima bab dengan sistematika penulisan sebagai berikut:

- **BAB I: Pendahuluan**
Bab ini meliputi latar belakang penelitian, perumusan masalah, tujuan penelitian, batasan masalah, metodologi, sistematika penulisan penelitian dan relevansi.
- **BAB II: Tinjauan Pustaka**
Bab ini menjelaskan tentang dasar teori yang dibutuhkan dalam pengerjaan penelitian ini, yang meliputi teori dasar citra uang, teori dasar sensor dan aktuator yang digunakan, teori dasar algoritma pengenalan citra uang, dan teori dasar mikrokontroler Arduino Mega.
- **BAB III: Perancangan Sistem**
Bab ini menjelaskan tentang perencanaan sistem yang meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk pembuatan alat ini.
- **BAB IV: Pengujian dan Analisis**

Bab ini menjelaskan tentang hasil yang didapat dari pengujian tiap blok sistem baik masing-masing maupun secara keseluruhan.

- **BAB V: Penutup**

Bab ini menjelaskan tentang kesimpulan dari pengujian sistem yang meliputi tingkat keberhasilan kerja sistem serta pemberian saran untuk dikembangkan selanjutnya.

1.7. Relevansi

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Dapat melakukan transaksi secara otomatis tanpa membutuhkan tenaga penjual manusia, dan transaksi dapat dilakukan dengan cepat dan akurat.
2. Dapat diintegrasikan dengan sistem penjualan barang *vending machine* maupun sistem lainnya yang membutuhkan sistem pembayaran yang terotomasi.

BAB II

TINJAUAN PUSTAKA

Suatu penelitian memerlukan teori-teori yang sudah ada sebelumnya untuk dikaji lebih dalam memperkuat argumen penulis. Teori tersebut digunakan untuk membantu penulis dan sebagai dasar dalam membuat suatu penelitian.

Pada bab ini terdapat teori dasar yang menjadi landasan untuk merumuskan dan menyelesaikan masalah yang akan dibahas pada penelitian ini. Pada bagian ini terdapat tinjauan pustaka tentang komponen yang akan digunakan untuk membuat alat pada penelitian ini.

2.1. Mesin Penjual Swalayan (*Vending Machine*)

Vending machine atau mesin swalayan adalah suatu mesin otomatis yang mampu menjual berbagai macam produk seperti makanan, minuman, rokok, koran, dan produk lainnya kepada konsumen menggunakan uang atau kartu untuk transaksi seperti kartu kredit atau kartu khusus untuk *vending machine*.

Vending machine melakukan transaksi dengan konsumen tanpa membutuhkan peran penjual manusia sehingga transaksi dilakukan sepenuhnya oleh mesin. Mekanisme *vending machine* terdiri dari sistem pembayaran, pengeluaran produk, dan beberapa mesin penjual swalayan memiliki sistem pengembalian uang kembalian. Jenis pembayaran yang digunakan beragam mulai dari uang koin, uang kertas, kartu debit/ kredit, dan sarana pembayaran lain seperti kupon atau kartu khusus. Pengembalian uang pada umumnya menggunakan uang koin tetapi beberapa *vending machine* juga mampu memberikan kembalian uang kertas.



Gambar 2.1 Contoh *vending machine*[2]

2.2. Uang Kertas Rupiah Tahun Emisi 2016

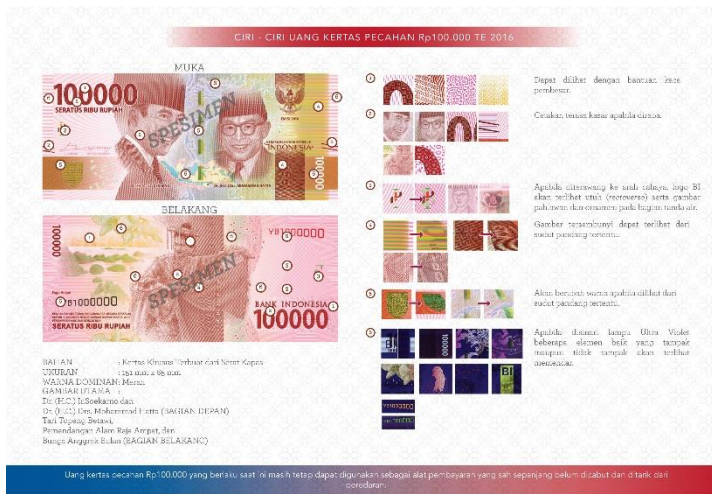
Rupiah adalah mata uang resmi Indonesia yang diedarkan oleh Bank Indonesia (BI). Pada tanggal 19 Desember 2016, uang Rupiah Tahun Emisi 2016 diresmikan. Pengeluaran dan pengedaran uang Rupiah TE 2016 merupakan salah satu pelaksanaan amanat UU Mata Uang. Pecahan mata uang kertas Rupiah yang baru adalah nominal Rp 100.000,00, Rp 50.000,00, Rp 20.000,00, Rp 10.000,00, Rp 5.000,00, Rp 2.000,00, Rp 1.000,00. Sedangkan untuk uang logam terdapat pecahan Rp 1.000,00, Rp 500,00, Rp 200,00, dan Rp 100,00.



Gambar 2.2 Pecahan uang kertas rupiah tahun emisi 2016/3/

Menurut Pasal 8 Undang-Undang Republik Indonesia Nomor 7 Tahun 2011 yang mengatur tentang desain mata uang Rupiah, desain mata uang rupiah harus memiliki ciri, tanda tertentu, ukuran, dan unsur pengaman[4]. Hal yang akan difokuskan dalam penelitian ini adalah unsur pengaman (*security features*) dari uang kertas rupiah. Selain untuk memastikan keaslian mata uang, unsur pengaman juga berfungsi sebagai pembeda antara satu pecahan dengan pecahan lainnya. Unsur pengaman dibagi menjadi dua, yaitu unsur pengaman terbuka dan tidak terbuka. Unsur pengaman terbuka cukup mudah untuk dideteksi oleh masyarakat umum, baik dengan teknik 3D (dilihat, diraba, diterawang), maupun dengan bantuan alat seperti sinar ultraviolet atau kaca pembesar. Sedangkan untuk unsur pengaman tertutup membutuhkan sensor khusus

yang hanya dimiliki pihak tertentu.



Gambar 2.3 Ciri-ciri dan unsur pengaman Rp 100.000,00[3]

2.3. Arduino Mega 2560

Arduino adalah perangkat elektronik *open-source* berbasis mikrokontroler yang mampu mengendalikan komponen elektronika dengan mudah. Mikrokontroler adalah sebuah komputer kecil dalam sebuah IC (*integrated circuit*), yang memiliki satu atau lebih CPU, memori, dan GPIO (*general purpose input output*) yang dapat diprogram. Arduino menggunakan bahasa pemrograman dan *library* yang didasari dari bahasa pemrograman C++ dan perangkat lunak Arduino (IDE).

Arduino Mega 2560 adalah jenis mikrokontroler Arduino yang memiliki pin I/O (*input/output*) yang lebih banyak daripada Arduino Uno. Arduino Mega 2560 menggunakan mikrokontroler ATmega2560, sebuah mikrokontroler CMOS 8-bit dengan daya rendah. Arduino Mega 2560 memiliki 54 pin *digital input/output*, 16 pin *analog input*, 4 UARTs (*hardware serial ports*), 16 MHz *crystal oscillator*, koneksi USB, *power jack*, ICSP *header*, dan tombol *reset*. Berikut ini spesifikasi lengkap dari Arduino Mega 2560:

- Mikrokontroler: ATmega2560
- Tegangan operasi: 5V

- Tegangan *input* (direkomendasikan): 7-12V
- Tegangan *input* (batas): 6-20V
- Pin *Digital I/O*: 54 pin (15 PWM *output*)
- Pin *Analog input*: 16 pin
- Arus DC untuk pin I/O: 20mA
- Arus DC untuk pin 3.3V: 50mA
- *Flash memory*: 256 KB, 8 KB digunakan oleh *bootloader*
- SRAM: 8 KB
- EEPROM: 4 KB
- Kecepatan *clock*: 16 MHz
- LED_BUILTIN: 13
- Panjang: 101.52 mm
- Lebar: 53.3 mm
- Berat: 37 gram



Gambar 2.4 Arduino Mega 2560

2.4. Sinar Ultraviolet (UV)

Ultraviolet (UV) adalah radiasi elektromagnetik dengan panjang gelombang berkisar antara 10 nm sampai 400 nm, lebih pendek dari sinar tampak, namun lebih panjang dari sinar X-ray. Spektrum elektromagnetik dari radiasi ultraviolet (UVR) berdasarkan ISO 21348 dapat dikategorikan berdasarkan rentang panjang gelombangnya (λ), yakni[5]:

- VUV: $10 \leq \lambda < 200$
- EUV: $10 \leq \lambda < 121$
- H Lyman- α : $121 \leq \lambda < 122$
- FUV: $122 \leq \lambda < 200$

- UVC: $100 \leq \lambda < 280$
- MUV: $200 \leq \lambda < 300$
- UVB: $280 \leq \lambda < 315$
- NUV: $300 \leq \lambda < 400$
- UVA: $315 \leq \lambda < 400$

Sinar ultraviolet yang digunakan pada tugas akhir ini adalah sinar ultraviolet yang bersumber dari *blacklight* yang memancarkan radiasi UVA dengan sedikit cahaya tampak. Sinar ultraviolet tersebut dapat memunculkan unsur pengaman tersembunyi yang terdapat pada cetakan uang.



Gambar 2.5 Lampu ultraviolet

2.5. Motor DC

Motor DC adalah sebuah mesin elektrik berputar yang mengubah energi listrik DC menjadi energi mekanik. Motor DC biasanya digunakan pada berbagai perangkat elektronik yang menghasilkan gerakan kinetik pada sistem kerjanya. Perputaran motor DC tergantung pada polaritas listrik DC yang diberikan, baik searah maupun berlawanan dengan jarum jam. Umumnya Motor DC berotasi dari 3000 rpm (*revolutions per minute*) sampai 8000 rpm dengan tegangan operasional 1.5 V – 24V.

Motor DC terdiri dari 2 bagian yaitu stator dan rotor. Sesuai dengan Namanya, stator adalah bagian dari motor yang statis atau tidak berputar. Stator terdiri dari rangka dan kumparan medan. Rotor adalah bagian motor yang berputar, terdiri dari kerangka magnet, kutub motor, kumparan medan magnet, kumparan jangkar, komutator, dan *brushes*.

Terdapat 3 jenis koneksi antara stator dan rotor pada motor DC yaitu seri, paralel, dan kompon (gabungan seri dan paralel). Setiap jenis memiliki karakteristiknya terhadap torsi bebannya[6].



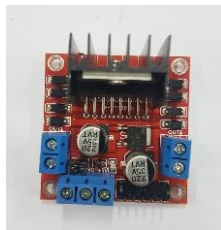
Gambar 2.6 Motor DC

2.6. Modul L298 DC Motor Driver

Modul L298 adalah modul *driver* motor DC yang umum digunakan untuk mengontrol motor DC. Modul L298 ini mampu mengontrol kecepatan serta arah perputaran motor DC. Modul L298 merupakan sebuah IC tipe H-bridge yang mampu mengendalikan beban induktif antara lain relay, solenoid, motor DC dan motor *stepper*. Arah putaran motor DC dapat diatur dengan mengatur *output* pin *enable* pada modul *driver*, sedangkan kecepatan putaran motor DC diatur melalui *input* sinyal PWM pada modul.

Spesifikasi dari modul *driver* motor DC L298 ini adalah sebagai berikut:

- Menggunakan IC L298N (Double H bridge Drive Chip)
- Tegangan minimal: 5-35 V
- Tegangan operasional: 5V
- Arus *input*: 0-35 mA
- Arus maksimal *output*: 2A
- Daya maksimal: 25 W
- Dimensi modul: 43 x 43 x 26 mm
- Berat: 26 g



Gambar 2.7 Modul L298

2.7. Motor Stepper

Motor *stepper* adalah sebuah motor DC *brushless* yang membagi satu putaran penuh motornya ke dalam beberapa “*step*”. Setiap masukan pulsa listrik menggerakkan rotor motor *stepper* sebanyak sebuah sudut diskrit yang disebut *step angle*[7]. Karena sistem kerjanya yang bergerak setiap *step*, posisi putaran motor *stepper* dapat diatur secara akurat meskipun bekerja sebagai sistem *open loop*, atau dengan kata lain tidak membutuhkan *feedback*. Karena keakuratannya, motor *stepper* biasanya digunakan pada banyak aplikasi di bidang industri.

Motor *stepper* memiliki beberapa elektromagnet bergigi yang diletakan di sekeliling roda gigi motor *stepper*. Ketika salah satu elektromagnet diberikan suplai listrik, secara magnetis ia akan menarik gigi roda motor. Ketika gigi roda terletak selaras dengan elektromagnet pertama, maka gigi roda motor akan sedikit menyimpang dari elektromagnet berikutnya. Maka dari itu, ketika elektromagnet berikutnya diaktifkan dan elektromagnet sebelumnya dinonaktifkan, roda gigi akan berputar sedikit untuk menyesuaikan dengan elektromagnet berikutnya. Setiap putaran tersebut disebut sebagai “*step*” dari motor *stepper*. Proses tersebut dilakukan terus menerus untuk menghasilkan perputaran motor dengan sudut yang akurat.



Gambar 2.8 Motor *stepper*

2.8. Modul ULN2003 Stepper Motor Driver

Modul ULN2003 adalah modul *driver stepper motor* yang digunakan untuk mengontrol motor *stepper* 4 fase dengan mudah. Terdapat 4 koneksi *input* untuk koneksi ke mikrokontroler, pin *power supply* untuk tegangan stepper motor, koneksi *header* untuk motor *stepper*, dan 4 LED untuk mengindikasikan kondisi *step* dari motor *stepper*[8].

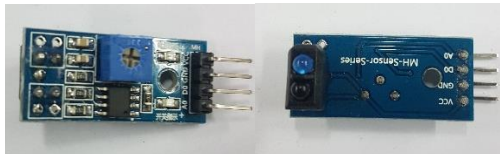


Gambar 2.9 Modul ULN2003

2.9. Modul Sensor *Infrared* TCRT5000

TCRT5000 adalah modul sensor yang memancarkan sinar inframerah (*infrared*) untuk mendeteksi keberadaan benda. Modul TCRT5000 tersusun dari *emitter* inframerah untuk memancarkan sinar inframerah dan *phototransistor* untuk menangkap pantulan sinar inframerah yang dipancarkan. Pada modul ini terdapat 4 pin yakni pin *power supply* dengan rentang kerja antara 3.3 – 5 volt, pin *ground*, 1 pin *analog output* yang mengeluarkan keluaran berupa tegangan analog yang merepresentasikan jarak antara benda terdeteksi dan *phototransistor*, dan 1 pin *digital output* yang mengeluarkan keluaran digital yang dapat diatur sedemikian rupa dengan mengatur sensitivitas sensor untuk menentukan keberadaan benda. Sensitivitas sensor dapat diatur dengan memutar potensiometer yang terdapat pada modul.

Emitter inframerah memancarkan sinar inframerah yang memiliki panjang gelombang 950 nm, yang tidak tampak oleh mata telanjang. *Phototransistor* juga dirancang dengan filter yang dapat menghalangi sinar tampak agar kinerjanya dapat terjaga meskipun dalam kondisi cahaya yang beragam. Modul sensor TCRT5000 dapat mendeteksi benda secara optimal pada rentang jarak 0.2 mm sampai 15 mm[9].



Gambar 2.10 Modul TCRT5000

2.10. Modul 2-Channel Relay

Modul relay adalah sebuah *switch* yang dapat dioperasikan secara elektrik untuk mengendalikan kondisi *on* maupun *off* dari *switch* tersebut. Pengendalian relay dapat dilakukan dengan memberikan sinyal tegangan *high* atau *low* dari mikrokontroler. Pada relay, tegangan rendah yang digunakan untuk mengendalikan modul relay tidak terhubung dengan tegangan tinggi yang dimaksudkan untuk terhubung dengan perangkat yang ingin dikendalikan, maka dari itu relay melindungi rangkaian satu dengan yang lainnya[10].

Setiap *channel* pada modul relay memiliki 3 koneksi yaitu NC (*normally closed*), COM, dan NO (*normally open*). Pada NC, rangkaian akan terhubung (*closed circuit*) pada kondisi normal dan akan menjadi terputus (*open circuit*) ketika *switch* diberikan *trigger*. Begitu pula sebaliknya pada NO, di mana rangkaian akan terbuka pada kondisi normal (*open circuit*) dan akan menjadi tertutup ketika *switch* diberikan *trigger*.



Gambar 2.11 Modul 2-channel Relay

2.11. LCD Display 16x2

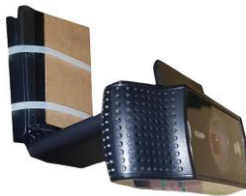
Layar LCD (*Liquid Crystal Display*) adalah komponen elektronika terpadu yang digunakan untuk menampilkan *output* berupa huruf maupun angka. LCD Display 16x2 adalah LCD yang mampu menampilkan 2 baris dari 16 karakter pada satu waktu. LCD biasanya digunakan untuk menampilkan variabel dalam suatu sistem elektronika. LCD memiliki 2 register, yaitu *command* dan *data*[11]. Register *command* adalah instruksi yang diberikan kepada LCD, sedangkan *data* adalah kode ASCII dari karakter yang ingin ditampilkan pada LCD.



Gambar 2.12 LCD display 16x2

2.12. Kamera USB

Kamera adalah instrumen optik yang digunakan untuk mengambil gambar maupun video. Kamera bekerja dengan menangkap spektrum cahaya tampak dengan sebagian dari spektrum elektromagnetik[12]. Citra analog yang ditangkap oleh kamera akan dijadikan informasi digital untuk ditampilkan secara digital. Pada tugas akhir ini, kamera yang digunakan adalah kamera dengan koneksi USB untuk mempermudah akuisisi dan pengolahan data citra uang. Selain itu, kamera USB yang digunakan juga berukuran kecil dan ringan sehingga mempermudah penempatan kamera pada sistem.



Gambar 2.13 Kamera USB

2.13. 4x4 Matrix Membrane Keypad

4x4 matrix membrane keypad adalah suatu komponen antarmuka (*interface*) elektronik yang digunakan untuk memberikan *input* digital kepada sistem dengan mudah. Keypad ini memiliki desain yang sangat tipis dengan ukuran 6,9 x 7,6 cm, tegangan kerja dengan tegangan maksimal 24 VDC, 8 pin digital untuk mengakses keypad, dan 16 pilihan tombol[13].



Gambar 2.14 4x4 matrix membrane keypad

2.14. Computer Vision

Computer vision adalah suatu bidang ilmu gabungan yang berhubungan dengan bagaimana komputer dapat memperoleh pemahaman tingkat tinggi dari citra digital maupun video. Dari perspektif ilmu teknik, *computer vision* bertujuan untuk mengotomatisasi tugas-tugas yang dapat dilakukan oleh sistem visual manusia[14]. *Computer vision* melingkupi metode akuisisi citra digital, pemrosesan citra digital, dan analisa citra digital. Citra digital adalah suatu representasi citra dalam bentuk angka.

Digital image acquisition atau akuisisi citra digital berhubungan dengan bagaimana komputer mampu memperoleh informasi yang membentuk suatu citra. *Digital image processing* atau pemrosesan citra digital adalah suatu penggunaan algoritma komputer yang digunakan untuk melakukan pemrosesan citra dalam citra digital[15]. *Digital image analysis* atau analisa citra digital adalah ekstraksi informasi yang berarti dari suatu citra digital[16].

2.15. Rekognisi Pola Citra

Bidang pengenalan pola berkaitan dengan pengenalan pola dalam data secara otomatis melalui penggunaan algoritma komputer. Rekognisi pola ini digunakan untuk mengambil tindakan seperti mengklasifikasikan data ke dalam berbagai kategori[17]. Rekognisi berdasarkan citra banyak digunakan dalam berbagai aplikasi, contohnya pengenalan identitas berdasarkan citra wajah, pengenalan nominal uang, pembuatan *self-driving car*, robot terautomasi, dan aplikasi lain yang membutuhkan kemampuan komputer untuk mengenali benda berdasarkan citra. Kemampuan komputer untuk melakukan rekognisi citra merupakan penerapan dari *machine learning* dan implementasi dari *computer vision*.

Komputer mampu mengenali benda dengan mengenali pola dari data citra. Data citra yang diakuisisi oleh komputer merupakan kumpulan angka-angka yang merepresentasikan nilai untuk setiap piksel dalam citra digital. Untuk melakukan rekognisi berdasarkan citra, komputer perlu dilakukan *training* terlebih dahulu di mana komputer akan menerima sampel-sampel citra yang mewakili suatu barang yang akan direkognisi.

2.16. *Principal Component Analysis (PCA)*

PCA (*Principal Component Analysis*) adalah suatu metode transformasi linear yang mendahulukan arah transformasi yang memaksimalkan variasi data. PCA ditemukan oleh Karl Pearson[18] dan dikembangkan oleh Harold Hotelling[19]. PCA merupakan teknik standar yang digunakan untuk memperkirakan data asli menggunakan matriks dengan dimensi yang lebih rendah[20]. PCA dilakukan dengan mencari *eigenvector* dari matriks kovarians suatu data.

Kovarians (*covariance*) adalah ukuran variabilitas gabungan dari dua variabel acak[21]. Dengan kata lain, kovarians adalah suatu ukuran yang menunjukkan tingkat kesesuaian dua variabel. Nilai kovarians positif menunjukkan bahwa variabel satu dengan yang lain bersesuaian ketika digerakkan ke arah yang sama, sedangkan sebaliknya nilai kovarians negatif menunjukkan bahwa ketika variabel yang satu digerakkan ke satu arah, variabel lainnya akan bergerak ke arah yang sebaliknya. Kovarians yang bernilai nol menunjukkan bahwa variabel satu dengan yang lain tidak memiliki relasi.

Matriks kovarians adalah sebuah matriks di mana elemen ke i dan j adalah kovarians antara elemen ke i dan j dari vektor acak. Matriks kovarians selalu berbentuk matriks persegi. Untuk memperoleh matriks kovarians dapat digunakan persamaan berikut[22]:

$$C = \frac{1}{n} \sum_{i=0}^n (x_i - x)(x_i - x)^T \quad (2.1)$$

Di mana C adalah matriks kovarians, n adalah jumlah sampel, dan x adalah rata-rata (mean) dari x_i .

Eigenvector adalah suatu vektor bukan nol yang melakukan transformasi linear dengan hanya mengubah faktor skalar tanpa mengubah arah vektor. *Eigenvalue* (λ) dan *eigenvector* (x) dari matriks kovarians (C) ditunjukkan dalam persamaan berikut[23]:

$$Cx = \lambda x; x \neq 0 \quad (2.2)$$

Dari persamaan di atas dapat diketahui bahwa matriks persegi C yang

ditransformasi oleh vektor bukan nol x (*eigenvector*) akan menghasilkan hasil yang sama dengan nilai λ (*eigenvalue*) yang dikalikan vektor x itu sendiri. Istilah “eigen” itu sendiri diambil dari Bahasa Jerman yang berarti ”layak” atau “karakteristik”[23], maka dari itu *eigenvector* atau vektor x adalah suatu “vektor karakteristik” dari matriks kovarians yang sesuai dengan nilai λ (*eigenvalue*). Untuk memperoleh *eigenvector* dari suatu matriks, pertama-tama nilai *eigenvalue* harus ditentukan terlebih dahulu. Nilai-nilai *eigenvalue* yang memenuhi persamaan (2.2) dapat diketahui dengan menurunkan persamaan (2.2). Persamaan (2.2) dapat ditulis kembali menjadi persamaan (2.3)[24].

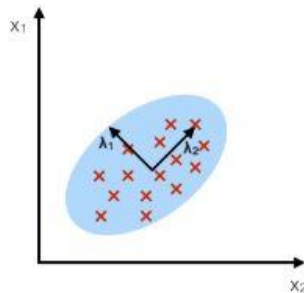
$$(C - \lambda I)x = 0 \quad (2.3)$$

Berdasarkan teorema Cramer, solusi dengan $x \neq 0$ dapat diperoleh jika dan hanya jika determinan koefisien persamaan (2.3) adalah nol[23]:

$$\det(C - \lambda I) = 0 \quad (2.4)$$

Di mana I adalah matriks identitas dan *eigenvalue* merupakan akar-akar dari persamaan determinan (2.4). *Eigenvalue* yang diperoleh dapat dikembalikan ke persamaan (2.2) untuk memperoleh *eigenvector* yang memenuhi.

Hasil *eigenvectors* yang diperoleh akan disusun menjadi matriks transformasi di mana data-data akan diproyeksikan dengan dimensi yang telah direduksi sekaligus mempertahankan variasi data.



Gambar 2.15 Ilustrasi PCA[25]

2.17. *Linear Discriminant Analysis (LDA)*

LDA (*Linear Discriminant Analysis*) adalah metode transformasi linear *supervised* yang mencari arah transformasi dengan memaksimalkan

separasi antara kelas yang berbeda. Berbeda dengan PCA, LDA menggunakan *class labels* dalam melakukan klasifikasi data., dengan kata lain, LDA mengelompokkan grup data dengan kelas yang sama dan memisahkan grup data lain yang tidak sekelas. LDA dilakukan dengan memperoleh *generalized eigenvectors* dari *scatter matrix* suatu kelompok data dalam matriks. LDA ditemukan oleh Ronald Fisher pada tahun 1936[26].

Scatter matrix adalah sebuah matriks yang menunjukkan tingkat penyebaran (*scatter*) dalam suatu kelompok data. Untuk menyelesaikan *generalized eigenvectors problem* dan memperoleh *generalized eigenvectors*, dibutuhkan *within class scatter matrix*, yaitu suatu matrix yang menghitung tingkat penyebaran antara kelompok data dalam satu kelas; dan *between class scatter matrix*, yaitu matrix yang menghitung tingkat penyebaran antar kelas. *Within class scatter matrix* (S_w) dapat ditentukan dengan persamaan berikut[24]:

$$S_i = \sum_{x \in X_i} (x - m_i)(x - m_i)^T \quad (2.5)$$

$$S_w = \sum_{i=1}^C S_i \quad (2.6)$$

Di mana C adalah jumlah kelas, S_i adalah *scatter matrix* untuk kelas i , dan m_i adalah rata-rata atau *mean* kelompok data dalam kelas i . Untuk memperoleh *between class scatter matrix* (S_B) dapat digunakan persamaan berikut[24]:

$$S_B = \sum_{i=1}^C n_i (m_i - m)(m_i - m)^T \quad (2.7)$$

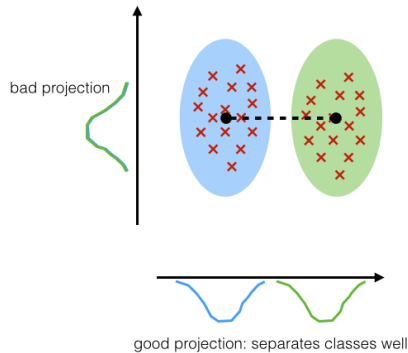
Di mana n_i adalah jumlah kelompok data dalam kelas i , m_i adalah *mean* dari kelompok data kelas i , dan m adalah *mean* dari seluruh kelompok data.

Setelah memperoleh kedua *scatter matrices* tersebut, dapat ditentukan *generalized eigenvectors* yang memenuhi *generalized eigenvalue problem* berikut[24]:

$$S_B v = \lambda S_w v \quad (2.8)$$

Di mana v adalah *generalized eigenvectors* dan λ adalah *generalized eigenvalues* dari kedua matriks S_B dan S_w . *Eigenvectors* yang diperoleh kemudian diurutkan dari yang terbesar sampai terkecil dengan maksimal

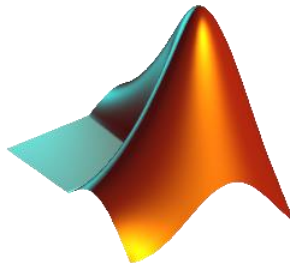
jumlah *eigenvectors* yang diambil sejumlah $(C-1)$. *Eigenvectors* tersebut menjadi vektor basis yang memaksimalkan separasi antar kelas kelompok data, di mana seluruh data akan diproyeksikan ke dalam vektor tersebut.



Gambar 2.16 Ilustrasi LDA[25/

2.18. MATLAB

MATLAB (*matrix laboratory*) merupakan *multi-paradigm numerical computing environment* dan *proprietary programming language* yang dikembangkan oleh MathWorks[27]. MATLAB mampu melakukan manipulasi matriks, membuat plot dari fungsi dan data, implementasi algoritma, pembuatan *user interface* (UI), dan membuat antarmuka dengan program yang ditulis menggunakan bahasa pemrograman lainnya, termasuk C, C++, C#, Java, Fortran, dan Python.



Gambar 2.17 Logo MATLAB[27/

2.19. Tinjauan Pustaka

Tinjauan pustaka bertujuan untuk membandingkan perangkat yang telah ada dan dikembangkan sebelumnya dengan perangkat yang dirancang pada tugas akhir ini. Berikut merupakan judul paper atau proek yang dibandingkan dengan proyek pada tugas akhir ini.

2.19.1. *Two-Dimensional Principal Component Analysis based Independent Component Analysis for face recognition*[28]

Sebelum menjalankan banyak algoritma pemrosesan gambar dan audio. Kemudian kita dapat menghapus data yang berlebihan dan menyimpan fitur yang berguna untuk analisis di masa mendatang. Independent Component Analysis adalah algoritma pengurangan dimensi yang terkenal. Algoritme biasanya dijalankan sebagai prosedur preprocessing untuk menjalankan Kompleksitas Independen sebelum menjalankan algoritma Analisis Komponen Independen. Kami mengusulkan analisis komponen utama dua dimensi berdasarkan algoritma analisis komponen independen, yang memproses gambar dua dimensi secara langsung dalam prosedur preprocessing. Eksperimen kontras pada database Yale membuktikan bahwa algoritma kami lebih efektif daripada algoritma PCA, 2dPCA, dan ICA klasik.

2.19.2. *Linear Principal Component Discriminant Analysis*[29]

Kami mengusulkan serangkaian metode analisis data untuk kedua teknik pembelajaran yang diawasi dan tidak diawasi. Tiga tujuan dari hubungan data dan karakteristik digunakan untuk membangun kerangka kerja seragam dari metode yang kami usulkan, yang terinspirasi oleh analisis komponen utama dan analisis diskriminan linier. Dengan menggunakan tiga tujuan dan beberapa kombinasi dari mereka, kami menyelidiki dan menggambarkan kinerja metode yang diusulkan. Kami menggunakan data simulasi dan data Iris klasik untuk menyelidiki metode yang diusulkan. Beberapa penemuan dan masalah dianalisis dan dibahas yang timbul dari hasil evaluasi. Keuntungan dari kerangka kerja yang diusulkan tidak hanya tergantung pada kemampuan penjelasan hubungan data, tetapi juga tergantung pada perpaduan beberapa teknik proyeksi data. Kami menyelidiki beberapa masalah penelitian potensial dari metode yang diusulkan. Beberapa karya yang memperluas studi saat ini dengan metode kernel dianalisis secara teoritis. Kami juga menyajikan beberapa karakteristik proposal dan mendiskusikan beberapa peluang terbuka dan pekerjaan di masa depan.

2.19.3. Automatic recognition and analysis of human faces and facial expression by LDA using wavelet transform[20]

Linear Discriminant Analysis (LDA) adalah salah satu teknik utama yang digunakan dalam sistem pengenalan wajah. Linear Discriminant Analysis (LDA) adalah skema terkenal untuk ekstraksi fitur dan pengurangan dimensi. Ini memberikan peningkatan kinerja selama metode standar pengenalan komponen utama (PCA) pengenalan wajah dengan memperkenalkan konsep kelas dan jarak antar kelas. Makalah ini memberikan ikhtisar PCA, berbagai varian LDA dan kelemahan dasarnya. Metode yang diusulkan mencakup pengembangan LDA klasik (yaitu LDA menggunakan pendekatan transformasi wavelet) yang meningkatkan kinerja seperti akurasi dan kompleksitas waktu. Eksperimen pada basis data wajah ORL dengan jelas menunjukkan hal ini dan perbandingan grafis dari algoritma dengan jelas menunjukkan tingkat pengakuan yang ditingkatkan dalam kasus algoritma yang diusulkan.

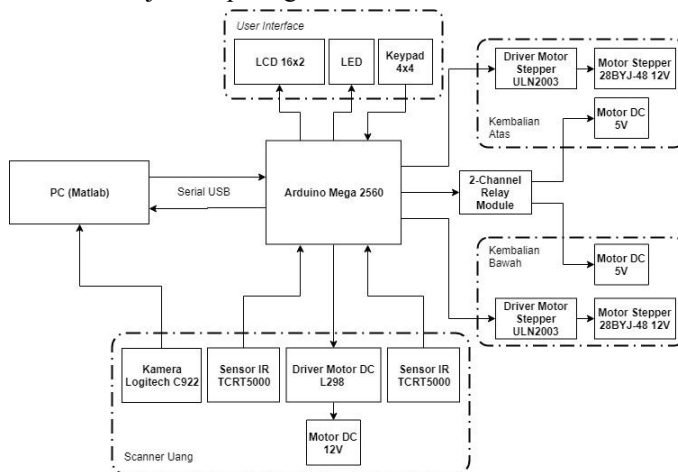
.....*Halaman ini sengaja dikosongkan*.....

BAB III PERANCANGAN SISTEM

Pada bab ini akan dijelaskan sistem yang dirancang secara keseluruhan, yakni perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat keras yang digunakan dan dirancang dalam sistem ini antara lain *scanner* uang yang terdiri dari mekanik *scanner*, lampu ultraviolet (UV), motor DC, *driver* motor DC L298, kamera USB, dan sensor IR TCRT5000; pengembalian uang yang terdiri dari mekanik sistem pengembalian, motor *stepper*, *driver* motor stepper ULN2003, motor DC, dan modul 2-channel relay; *microcontroller development board* Arduino Mega 2650, PC (*personal computer*), LCD 16x2, LED indikasi *output*, *keypad*, rangkaian suplai tegangan DC, dan bodi luar sistem. Perangkat lunak yang akan diimplementasikan pada sistem ini antara lain akuisisi citra uang, pemrosesan citra digital uang, analisa citra digital uang menggunakan PCA dan LDA, kalkulasi *distance*, dan program sistem pembayaran *vending machine*.

3.1. Diagram Blok Sistem

Diagram blok sistem pembayaran mesin penjual swalayan secara keseluruhan ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram blok sistem.

Pada tugas akhir ini, digunakan Arduino Mega 2560 sebagai pengendali sistem secara keseluruhan, sedangkan PC pada sistem digunakan sebagai pengolah data citra uang dengan perangkat lunak MATLAB. *Scanner* uang terdiri dari lampu ultraviolet, 2 sensor IR TCRT5000, *driver* motor DC L298, dan motor DC 12V. Pengembalian uang terdiri dari *driver* motor stepper ULN2003, motor stepper 28BYJ-48 12V, modul 2-channel relay, dan motor DC 5V. Perangkat keras yang digunakan dalam sistem ini meliputi:

- *Scanner* uang
Scanner uang terdiri dari mekanik *scanner*, kamera USB, lampu ultraviolet, 2 sensor IR TCRT5000, *driver* motor L298, dan motor DC 12V. Seluruh perangkat keras *scanner* uang berfungsi untuk mendeteksi dan menerima uang, akusisi citra digital uang, mendorong uang masuk, maupun mendorong uang kembali keluar.
- Pengembalian uang
 Pengembalian uang terdiri dari mekanik pengembalian uang, *driver* motor *stepper* ULN2003, motor *stepper* 28BYJ-48 12V, modul relay, dan motor DC 5V. Pengembalian uang berfungsi untuk memberikan kembalian sesuai hasil transaksi. Pada sistem ini terdapat 2 pengembalian pecahan uang, yaitu Rp 50.000,00 dan Rp 10.000,00.
- *User interface* (UI)
 Untuk berkomunikasi dengan sistem dengan mudah, diperlukan suatu *user interface* (UI) pada sistem. *User interface* terdiri dari LCD 16x2 dan LED untuk menampilkan keluaran sistem dan *keypad* 4x4 untuk melakukan perintah masukan kepada sistem.
- Rangkaian suplai tegangan DC
 Rangkaian suplai tegangan DC berfungsi sebagai catu daya tegangan DC 5V dan 12V pada seluruh komponen elektronik sistem. Rangkaian terdiri dari adapter DC 12V dan IC *voltage regulator* LM7805.
- Arduino Mega 2560
 Sistem secara keseluruhan dikendalikan oleh Arduino Mega 2560 yang menggunakan mikrokontroler ATmega2560. Arduino Mega berperan dalam menerima hasil deteksi sensor IR, berkomunikasi dengan PC secara serial, menerima perintah berdasarkan hasil pengolahan data oleh PC secara serial, menyalakan dan mengendalikan kecepatan dan arah putaran

motor DC *scanner* uang, mengendalikan motor stepper pengembalian uang, menyalakan motor DC pengembalian melalui relay, menampilkan hasil pengolahan data dan transaksi pada LCD, menerima perintah dari *keypad*, dan menunjukkan hasil transaksi pada LED.

- PC (*Personal Computer*)

PC berperan sebagai pengolah data *input* citra uang dengan perangkat lunak MATLAB. PC juga melakukan komunikasi dengan Arduino Mega secara serial untuk melakukan akuisisi citra dan mengirimkan informasi hasil deteksi citra uang. *Database* citra *training* juga tersimpan di dalam PC.

- Bodi luar sistem

Agar pengambilan citra uang tidak terpengaruh oleh kondisi pencahayaan dari luar, maka dibutuhkan bodi sistem yang mampu menutup bagian dalam sistem dan mencegah cahaya masuk yang berpotensi mengganggu deteksi citra uang.

3.2. Perancangan Perangkat Keras

Perangkat keras pada tugas akhir ini meliputi *scanner* uang, pengembalian uang, *user interface* (UI) sistem, rangkaian suplai tegangan DC, Arduino Mega 2560, dan bodi luar sistem.

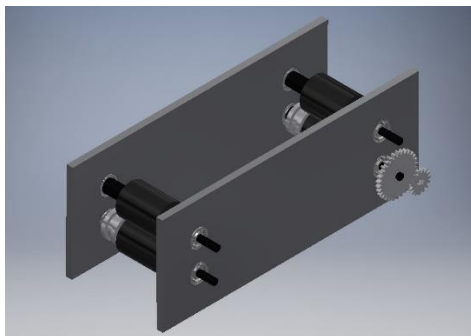
3.2.1. *Scanner* Uang

Scanner uang dimaksudkan sebagai komponen sistem yang bertugas untuk menerima uang secara otomatis, melakukan akuisisi citra digital uang yang diterima, dan mendorong uang baik masuk maupun keluar. *Scanner* uang meliputi mekanik *scanner*, *driver* motor DC L298, motor DC 12V, sensor IR TCRT2000, kamera USB, dan lampu ultraviolet.

3.2.1.1 Mekanik *Scanner* Uang

Mekanik *scanner* uang yang digunakan pada sistem ditunjukkan pada gambar 3.2. Rancangan mekanik dirancang menggunakan perangkat lunak Autodesk Inventor Professional 2019. Realisasi mekanik merupakan gabungan dari *cutting* laser akrilik dan 3D *printing*. Pada mekanik terdapat beberapa komponen mekanis yang disusun sedemikian rupa agar mekanik mampu menerima uang dan mendorong uang ke kedua arah. Komponen mekanis yang dimaksud tersebut terdiri dari bodi mekanik yang terbuat dari bahan akrilik, 4 poros dengan diameter 5 mm

yang terbuat dari filamen PLA hasil 3D *printing*, 4 buah karet pendorong kertas dengan diameter 25 mm, *bearing* 625Z dengan diameter dalam 5 mm, *timing belt* GT2MX dengan jumlah 320 gigi, *pulley* 20 gigi dengan diameter dalam 8 mm, dan *gear* motor pada salah satu poros.



Gambar 3.2 Rancangan mekanik *scanner*

Bodi mekanik berfungsi sebagai tempat peletakan seluruh komponen mekanik, *bearing* digunakan untuk meminimalkan gesekan antara poros dengan bodi mekanik agar beban torsi motor minimal, dan *pulley* sebagai peletakan *timing belt* yang bertujuan untuk mengatur keselarasan putaran mekanik dalam menerima uang kertas agar posisi uang kertas dalam *scanner* simetris. Meskipun motor DC yang digunakan dalam *scanner* bertegangan 12V, namun mengingat beban torsi yang harus diangkat oleh motor DC cukup besar, maka digunakan *gear* motor untuk menambah torsi motor DC.

3.2.1.2 *Driver Motor DC L298*

Driver motor DC L298 adalah komponen elektronik yang digunakan untuk mempermudah pengendalian arah dan kecepatan putar motor DC yang memutar poros mekanik *scanner*. *Driver* motor L298 disuplai oleh tegangan 12V yang diperoleh dari adapter DC 12V. Pada sistem ini, *driver* motor bertugas untuk mengatur arah putar motor DC pada *scanner* sesuai hasil pengenalan citra uang maupun *input* manual dari *user*. Pengaturan kecepatan hanya digunakan untuk mengatur kecepatan putar motor DC dengan konstan tanpa ada perubahan kecepatan di dalam sistem. Pin *enable* B merupakan pin *analog input* untuk menerima sinyal PWM yang mengatur kecepatan motor, pin *input* 3 dan 4 merupakan *pin* digital untuk

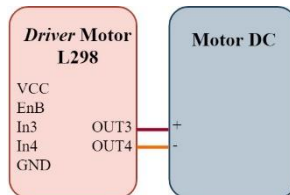
mengatur arah putar motor. Antarmuka *driver* motor L298 dengan Arduino Mega dapat dilihat pada gambar 3.3.



Gambar 3.3 Antarmuka *driver* motor L298 dengan Arduino Mega

3.2.1.3 Motor DC 12V

Motor DC yang digunakan untuk memutar mekanik *scanner* memiliki tegangan operasi sebesar 12V. Sistem menggunakan motor DC 12V karena dibutuhkan torsi motor yang cukup besar untuk menggerakkan seluruh mekanik *scanner*. Kecepatan dan arah putar motor DC dikendalikan oleh Arduino Mega melalui *driver* motor L298. Pada motor DC hanya terdapat 2 pin, yaitu pin positif dan negatif yang terhubung dengan pin *output* dari *driver* motor L298. Antarmuka antara motor DC dengan *driver* motor L298 ditunjukkan pada gambar 3.4.

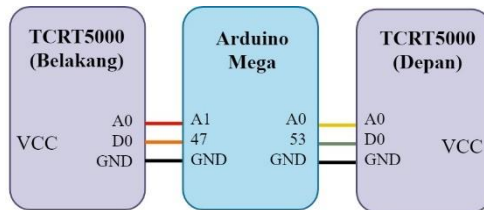


Gambar 3.4 Antarmuka *driver* motor L298 dengan motor DC

3.2.1.4 Sensor IR TCRT2000

Pada *scanner* uang yang dirancang terdapat 2 buah sensor IR TCRT2000 yang terletak pada 2 posisi yang berbeda. Sensor IR diletakkan pada mulut *scanner* yang berfungsi untuk mendeteksi keberadaan uang yang akan memutar motor DC dalam waktu tertentu, dan sensor IR lainnya diletakkan di ujung belakang *scanner* untuk mendeteksi keberadaan uang pada *scanner* dan menentukan apakah Arduino dapat melakukan komunikasi serial dengan PC untuk mengambil citra dengan

kamera. Sensor IR TCRT2000 terdiri dari pin VCC, pin GND, pin A0 sebagai *output* analog yang mendeteksi jarak antara sensor dan benda dalam besaran analog, dan pin D0 sebagai *output* digital yang mendeteksi keberadaan benda sesuai sensitivitas sensor yang ditentukan. Antarmuka dari kedua sensor IR yang digunakan dapat dilihat di gambar 3.5.



Gambar 3.5 Antarmuka TCRT5000 dengan Arduino Mega

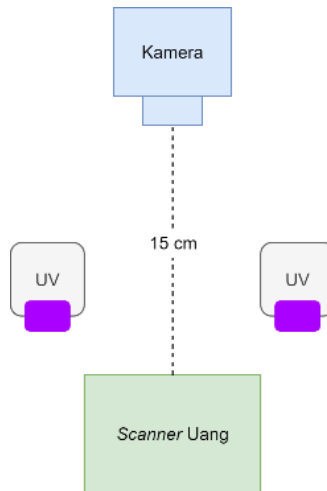
3.2.1.5 Kamera USB

Kamera USB pada sistem ini berperan sebagai *image sensor* yang bertugas untuk melakukan akuisisi citra digital yang masuk ke dalam *scanner*. Kamera USB yang digunakan pada tugas akhir ini adalah Logitech C922 Pro Stream 1080p. Kamera ini mampu mengambil gambar maupun video dengan resolusi maksimal 1080p atau *full HD*. Kemampuan mengambil gambar dengan resolusi tinggi dibutuhkan agar akuisisi citra dan analisa citra dapat dilakukan dengan lebih akurat. Kamera diletakkan 15 cm di atas mekanik *scanner* yang dengan sudut kemiringan 90° agar kamera dapat menangkap citra dengan kualitas gambar sebaik mungkin. Kamera USB dihubungkan langsung ke PC melalui USB *port* dan dikendalikan oleh perangkat lunak MATLAB. Posisi kamera pada sistem dapat dilihat pada gambar 3.6.

3.2.1.6 Lampu Ultraviolet

Lampu ultraviolet digunakan untuk mendeteksi keaslian uang rupiah. Di bawah sinar ultraviolet, unsur pengaman uang yang dicetak dengan tinta UV akan terlihat secara kasat mata maupun di hadapan kamera. Lampu ultraviolet yang digunakan merupakan lampu ultraviolet ROYALUX dengan *rating* tegangan 170-240 volt AC dan daya sebesar 4 watt. Pada sistem ini digunakan 2 buah lampu ultraviolet untuk memaksimalkan pancaran sinar ultraviolet terhadap uang. Lampu ultraviolet diletakkan di bagian sisi atas *scanner* yang dengan posisi lampu 90° menghadap ke bawah. Posisi kedua lampu ultraviolet

digambarkan pada gambar 3.6.



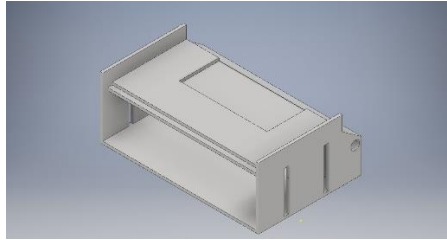
Gambar 3.6 Tampak depan pemasangan kamera dan lampu UV

3.2.2. Pengembalian Uang

Pada tugas akhir ini, pengembalian uang dilakukan dengan kombinasi antara mekanik pengembalian uang, motor *stepper*, relay, dan motor DC 5V. Sistem memberikan kembalian uang secara satu-persatu dengan mengandalkan putaran presisi dari motor *stepper*. Pada tugas akhir ini dirancang 2 buah sistem pengembalian uang dengan masing-masing pecahan sebesar Rp 10.000,00 dan Rp 50.000,00.

3.2.2.1. Mekanik Pengembalian Uang

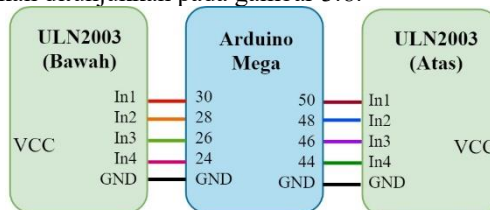
Sistem pada tugas akhir ini menggunakan mekanisme yang sederhana untuk mengembalikan uang. Tumpukan uang dengan pecahan nominal yang sama diletakkan pada mekanik uang, di mana terdapat suatu piringan persegi panjang yang diberikan tekanan ke atas oleh pegas. Ketika 1 lebar uang ingin dikeluarkan, motor *stepper* akan diputar sebanyak 360 derajat yang akan memutar pendorong uang kertas. Uang kertas yang terdorong akan masuk ke dalam dua pemutar yang diposisikan saling bersentuhan yang dihubungkan dengan motor DC 5V. Dua pemutar ini berfungsi untuk mengeluarkan uang kembalian tersebut. Rancangan secara kasar ditunjukkan pada gambar 3.7.



Gambar 3.7 Rancangan pengembalian uang sederhana

3.2.2.2. *Driver ULN2003*

Driver ULN2003 digunakan untuk mempermudah pengendalian motor stepper yang digunakan dalam pengembalian uang. *Driver* disuplai oleh tegangan 12V untuk memaksimalkan torsi motor stepper yang dihasilkan. Pin *logic* dari *driver* dihubungkan dengan pin digital pada Arduino mega. Karena pada tugas akhir ini akan diimplementasikan sistem dengan dua pecahan kembalian, maka digunakan 2 buah *driver* ULN2003 untuk 2 motor stepper. Antarmuka antara 2 buah *driver* motor yang digunakan ditunjukkan pada gambar 3.8.

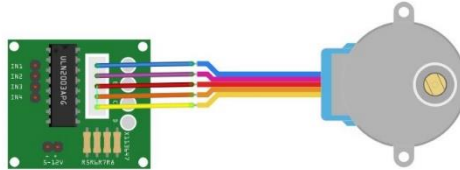


Gambar 3.8 Antarmuka *driver* ULN2003 dengan Arduino Mega

3.2.2.3. *Motor Stepper 28BYJ-48 12V*

Motor *stepper* 28BYJ-48 12V adalah motor *stepper* 4 fase dengan *rating* tegangan 12V. Motor *stepper* ini memiliki sudut *step* 5.625° per *step* atau dengan kata lain untuk berputar 1 putaran penuh (360°) dibutuhkan 64 *step*. Motor *stepper* berfungsi untuk memutar poros dengan karet pendorong uang pada mekanik pengembalian untuk mendorong uang keluar. Untuk mempermudah rangkaian elektronik maupun pengendalian motor, maka motor *stepper* dihubungkan dengan *driver* motor *stepper* ULN2003. Terdapat 5 pin *input* pada motor *stepper* 28BYJ-48 12V, yakni 4 pin yang mewakili masing-masing fase, dan 1 pin *ground*. Kelima pin tersebut dihubungkan dengan pin *output* dan *ground*

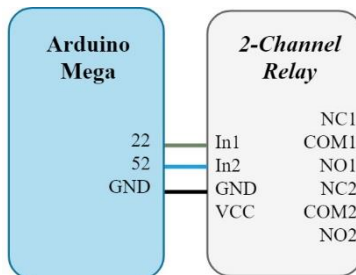
yang sudah disatukan pada motor stepper ULN2003. Koneksi rangkaian antara motor *stepper* 28BYJ-48 12V dan *driver* ULN2003 digambarkan pada gambar 3.9.



Gambar 3.9 Rangkaian antara motor stepper dan ULN2003

3.2.2.4. Modul 2-Channel Relay

Pada sistem ini modul relay digunakan untuk mengontrol keadaan aktif/tidaknya motor DC sekaligus mencegah *drop* tegangan pada Arduino Mega jika motor DC langsung dihubungkan dengan pin digital Arduino. Pada relay terdapat 3 pin yang terhubung dengan *switch* pada relay, yaitu pin *Normally Close* (NC), *Common* (COM), dan *Normally Open* (NO). Dua pin *input logic* relay terhubung dengan Arduino untuk mengontrol kondisi *switch* pada relay sedangkan relay disuplai langsung oleh rangkaian suplai tegangan DC; maka dari itu digunakan antarmuka seperti yang ditunjukkan pada gambar 3.10.

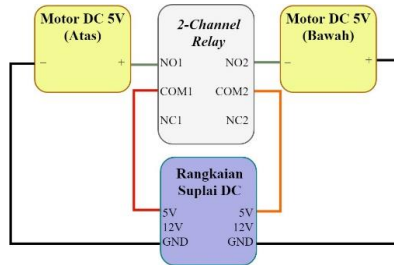


Gambar 3.10 Antarmuka 2-channel relay dengan Arduino Mega

3.2.2.5. Motor DC

Motor DC berfungsi untuk memutar mekanik penerus uang yang terdiri dari 2 buah poros dengan karet pada sekeliling poros. Penerus uang ini berfungsi untuk menarik uang yang terdorong keluar dari mekanik pengembalian dan meneruskannya ke luar. Dalam pengendalian motor

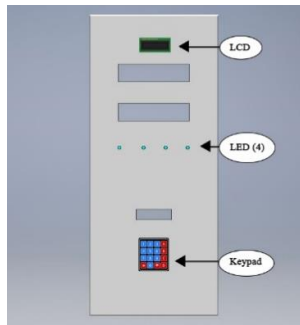
DC pada pengembalian uang ini digunakan *switch* pada relay untuk mencegah drop tegangan pada Arduino Mega. Penggunaan relay dilakukan dengan menghubungkan salah satu pin motor DC ke pin NO relay yang diteruskan melalui pin COM ke pin 5V dari rangkaian suplai tegangan DC, dan pin lainnya ke GND. Penentuan pin positif dan negatif motor ditentukan sesuai arah putaran motor yang diinginkan. Antarmuka antara komponen-komponen tersebut ditunjukkan pada gambar 3.11.



Gambar 3.11 Antarmuka motor DC, relay, dan rangkaian suplai

3.2.3. User Interface

Untuk mempermudah komunikasi antara pengguna (*user*) dengan sistem, dibutuhkan suatu *user interface*. Komunikasi yang dijematani antara *user* dan sistem pada tugas akhir ini meliputi penunjukan *input* dan *output* hasil deteksi uang, saldo tersisa, dan kondisi sistem menggunakan karakter angka maupun huruf pada LCD, pemberian perintah dari *user* kepada sistem melalui *keypad* 4x4, dan indikasi *output* transaksi pada rangkaian LED.

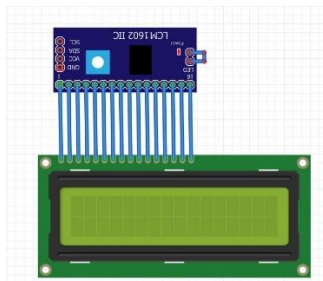


Gambar 3.12 Rancangan *user interface* sistem

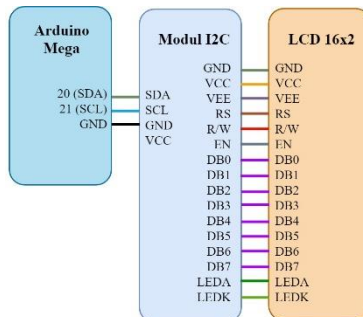
3.2.3.1 *Liquid Crystal Display 16x2 (LCD)*

LCD 16x2 pada UI sistem ini berfungsi untuk menampilkan hasil *input* maupun *output* yang dilakukan pada sistem Hasil yang dimaksud meliputi mode sistem yang sedang berlangsung, hasil deteksi uang, saldo yang tersisa, berhasil atau gagalnya transaksi, berhasil atau gagalnya pembacaan uang, sisa kembalian, dan hasil *input* perintah yang diberikan.

Untuk mempermudah perangkaian dan pemrograman LCD, digunakan modul I2C untuk LCD16x2. Modul I2C menggantikan koneksi pin *command* dan *data* dari LCD dengan komunikasi I2C menggunakan pin SDA dan SCL. Dengan menggunakan modul I2C, penggunaan pin pada mikrokontroler dapat dikurangi dari 16 pin menjadi hanya 4 pin yang terdiri dari VCC, GND, SDA, dan SCL. Pin SDA dan SCL dari modul I2C dihubungkan dengan SCA dan SCL dari Arduino Mega 2560 yang terletak pada pin 20 dan 21. Antarmuka antara LCD dan Arduino Mega ditunjukkan pada gambar 3.14.



Gambar 3.13 Rangkaian antara LCD dengan modul I2C



Gambar 3.14 Antarmuka LCD dengan Arduino Mega

3.2.3.2 Keypad 4x4

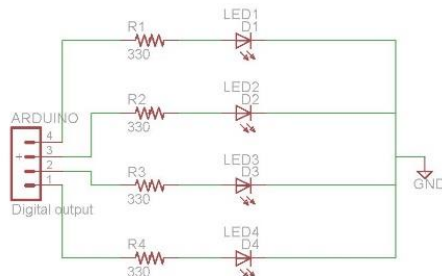
Untuk memberikan *input* perintah pada sistem seperti pemilihan mode antara mode deteksi dan mode normal, perintah pembatalan transaksi, perintah pengeluaran uang, pemilihan pilihan harga yang diinginkan, dan pemberian kembalian. *Keypad* 4x4 memiliki 8 pin yang mewakili 4 baris dan 4 kolom pada *keypad*. Pin *keypad* dihubungkan pada 8 pin *digital* Arduino Mega. Antarmuka antara *keypad* dan Arduino Mega dapat dilihat pada gambar 3.15.



Gambar 3.15 Antarmuka *keypad* dengan Arduino Mega

3.2.3.3 Rangkaian LED

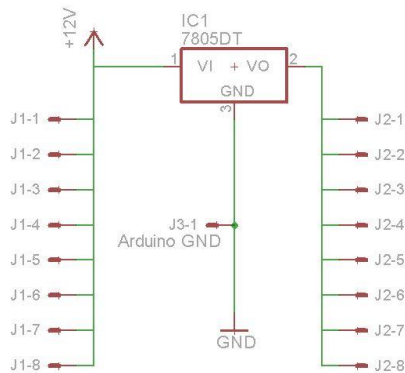
Karena tugas akhir ini dibatasi pada sistem pembayaran tanpa mekanik pengeluaran barang, rangkaian LED berfungsi untuk mewakili keluarnya barang pilihan dari hasil transaksi yang dilakukan. Pada sistem ini terdapat 4 buah LED yang mewakili 4 pilihan harga barang, yaitu Rp 10.000,00, Rp 20.000,00, Rp 50.000,00, dan Rp 100.000,00. LED terhubung dengan resistor 330 Ω dan 4 pin digital Arduino Mega. Skematik dari rangkaian LED dapat ditunjukkan pada gambar 3.16.



Gambar 3.16 Skematik rangkaian LED

3.2.4. Rangkaian Suplai Tegangan DC

Rangkaian suplai tegangan DC berfungsi untuk memberikan tegangan suplai kepada seluruh komponen elektronik pada sistem. Rangkaian terdiri dari DC *barrel power jack* untuk dihubungkan dengan adapter DC 12V, IC *voltage regulator* 5V LM7805 untuk menurunkan tegangan, dan pin *header male*. Rangkaian suplai tegangan DC menyediakan tegangan *output* sebesar 5V dan 12V untuk memenuhi kebutuhan tegangan suplai pada sistem tugas akhir ini. Skematik rangkaian suplai tegangan DC ditunjukkan pada gambar 3.17.



Gambar 3.17 Skematik rangkaian suplai tegangan DC.

3.2.5. Arduino Mega 2560

Arduino Mega 2560 berfungsi untuk mengontrol seluruh komponen elektronik pada sistem dengan menjalankan program sistem yang telah diunggah. Selain itu Arduino juga melakukan komunikasi secara serial dengan PC untuk mengirimkan perintah pengambilan citra uang dan menerima hasil deteksi citra uang.

Pinout Arduino Mega 2560 ditunjukkan pada gambar 3.18 dan pembagian *pinout* dari Arduino Mega 2560 terhadap komponen elektronik yang digunakan dalam tugas akhir ini dijelaskan pada tabel 3.1.

26	<i>Driver ULN2003 (bawah)</i>	In3
28	<i>Driver ULN2003 (bawah)</i>	In2
30	<i>Driver ULN2003 (bawah)</i>	In1
36	LED (1)	Anoda (+)
38	LED (2)	Anoda (+)
40	LED (3)	Anoda (+)
42	LED (4)	Anoda (+)
44	<i>Driver ULN2003 (atas)</i>	In4
46	<i>Driver ULN2003 (atas)</i>	In3
47	Sensor IR (belakang)	Digital <i>read</i>
48	<i>Driver ULN2003 (atas)</i>	In2
49	<i>Driver L298</i>	In3
50	<i>Driver ULN2003 (atas)</i>	In1
51	<i>Driver L298</i>	In4
52	Relay	In2
53	Sensor IR (depan)	Digital <i>read</i>

Tabel 3.1 Pembagian pin Arduino Mega 2560 pada sistem

3.2.6. PC (*Personal Computer*)

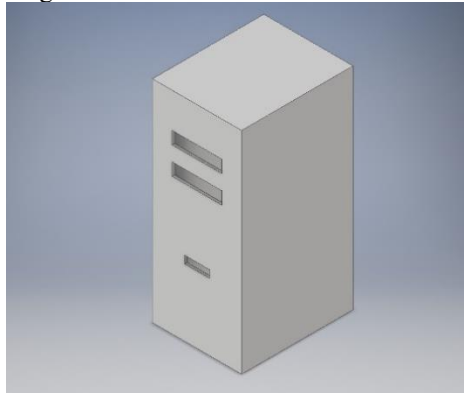
Pada tugas akhir ini, perangkat lunak MATLAB yang bertugas untuk melakukan akuisisi, proses, dan analisa citra uang dijalankan pada perangkat keras berupa laptop (PC). Laptop dihubungkan dengan Arduino Mega 2560 melalui kabel USB untuk melakukan komunikasi secara serial. Spesifikasi laptop yang digunakan dalam tugas akhir ini dijelaskan pada tabel 3.2.

Processor	Intel® Core™ i7-8750H CPU @ 2.20GHz
Memory	16 GB RAM
DirectX	DirectX 12
OS	Windows 10 Home 64-bit (10.0, Build 17134)
System Model	ROG GU510GM

Tabel 3.2 Spesifikasi PC yang digunakan pada sistem

3.2.7. Bodi Luar Sistem

Untuk mencegah gangguan dalam proses deteksi citra uang pada sistem dalam tugas akhir ini, maka pengambilan citra uang harus terhindar dari interferensi cahaya yang tidak diinginkan. Maka dari itu dibutuhkan suatu bodi luar yang mampu menghalangi cahaya luar agar tidak mengganggu kinerja sistem. Selain itu, bodi luar juga menjadi tempat penempatan seluruh komponen dalam sistem maupun *user interface* sistem. Bodi luar pada tugas akhir ini terbuat dari gabungan antara kayu MDF dan *duplex*. Pada tampak depan bodi luar sistem terdapat 3 lubang persegi panjang untuk penempatan masuknya uang ke *scanner* uang dan keluarnya uang untuk 2 pengembalian uang. Rancangan bodi luar sistem memiliki dimensi 35x45x85 cm. Desain rancangan dari bodi luar sistem ditunjukkan pada gambar 3.19.

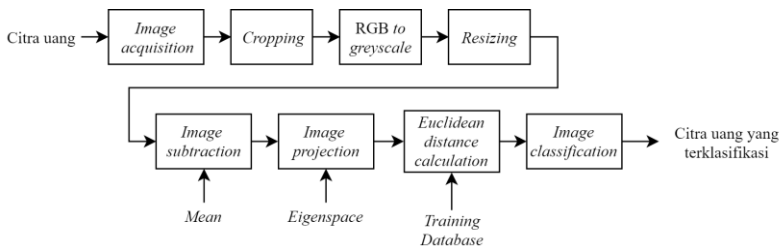


Gambar 3.19 Rancangan bodi luar sistem

3.3. Perancangan Perangkat Lunak

Perangkat lunak yang diimplementasikan pada tugas akhir ini meliputi program pada MATLAB yang terdiri dari program akuisisi citra uang, pemrosesan citra digital uang, penerapan LDA (*linear discriminant analysis*), kalkulasi *distance*, dan program pada Arduino Mega 2560 yang meliputi sistem pembayaran *vending machine*.

Program pengenalan citra uang meliputi dua tahap, yaitu penyusunan *database* citra latihan berdasarkan metode PCA maupun LDA, dan klasifikasi citra uang dengan kalkulasi jarak *Euclidean*. Seluruh langkah-langkah klasifikasi citra uang berdasarkan nominalnya dijelaskan pada diagram blok gambar 3.20.



Gambar 3.20 Diagram blok klasifikasi citra uang

Citra uang yang masuk ke dalam *scanner* uang akan di akuisisi oleh kamera dengan pengaturan kamera yang disesuaikan agar diperoleh citra uang yang jelas. Kemudian dilanjutkan dengan *cropping* di mana perangkat lunak hanya akan mengambil bagian citra yang penting dan membuang bagian yang tidak dibutuhkan. Proses *cropping* dilakukan secara otomatis yang disesuaikan dengan struktur perangkat keras. Citra yang sudah dipotong akan diubah dari komposisi warna RGB menjadi *greyscale*. Citra kemudian dikecilkan ukurannya pada tahap *resizing* untuk menyederhanakan komputasi. Citra yang telah diproses kemudian dikurangi oleh *mean* yang diperoleh dari tahap latihan, di mana *mean* merupakan rata-rata dari setiap citra latihan. Kemudian citra diproyeksikan ke dalam *subspace* PCA maupun LDA dengan mengalikan matriks citra dengan matriks *eigenspace* yang diperoleh dari proses latihan. Citra dalam ruang baru akan diukur jaraknya dengan seluruh basis data citra latihan dengan menyusun suatu matriks jarak yang mewakili jarak citra uji dengan masing-masing citra latihan. Klasifikasi citra berdasarkan nominalnya dapat dilakukan dengan mencari posisi baris dari jarak *Euclidean* terkecil pada matriks jarak. Keluaran dari sistem ini adalah citra uang masukan yang terklasifikasi berdasarkan nominalnya.

3.2.1. Akuisisi Citra Uang

Program akuisisi citra uang dilakukan melalui perangkat lunak MATLAB. akuisisi dilakukan oleh kamera Logitech C922 yang dihubungkan ke PC untuk dikendalikan oleh MATLAB. Pertama-tama, dilakukan program untuk mengakses dan mengatur properti/ pengaturan kamera yang diinginkan. Properti kamera yang diakses dan dikendalikan pada sistem ini meliputi fokus, eksposur, kompensasi lampu latar, keseimbangan putih, dan *gain*. Program yang digunakan untuk mengakses dan mengatur kamera adalah sebagai berikut:

```

cam = webcam(1);
cam.FocusMode = 'manual'
cam.Focus = 15;
cam.Exposure = -4;
cam.BacklightCompensation = 1;
cam.WhiteBalanceMode = 'manual'
cam.WhiteBalance = 4000;
cam.Gain = 255;

```

Selanjutnya, dilakukan pengambilan gambar ketika PC menerima perintah serial dari Arduino ketika sensor mendeteksi adanya uang. Citra digital uang yang diperoleh akan disimpan pada direktori yang sesuai. Program yang melakukan fungsi tersebut adalah sebagai berikut:

```

img = snapshot(cam);

```

3.2.2. Pemrosesan Citra Digital Uang

Pemrosesan citra digital uang dilakukan untuk memproses citra digital uang agar menjadi sesuai seperti kriteria citra yang diinginkan pada proses analisa citra. Pemrosesan citra digital uang pada tugas akhir ini meliputi 3 tahap, yaitu *cropping*, konversi *color space grayscale*, dan *resizing*.

Proses *cropping* dilakukan agar daerah citra yang diambil sesuai dengan yang diinginkan sekaligus membuang bagian citra yang tidak dibutuhkan. Ukuran *cropping* diperoleh dari hasil *trial* dan *error* sehingga diperoleh proses *crop* dengan pengambilan citra dari koordinat awal $x=55$ ditarik sepanjang 520 piksel dan koordinat awal $y=90$ ditarik sepanjang 240 piksel. Hasil proses *cropping* berupa citra digital berukuran 520x240 piksel. Program untuk menjalankan proses *cropping* adalah sebagai berikut:

```

img_crop = imcrop(img,[55 90 520 240]);

```

Kemudian dilanjutkan dengan proses pengubahan *color space* citra digital dari RGB menjadi *grayscale*. Hal ini dilakukan karena pada tugas akhir ini, algoritma yang digunakan bekerja dengan menganalisa data matriks citra dalam bentuk *grayscale*. Analisa citra dilakukan dalam *color space greyscale* bertujuan untuk menyederhanakan dimensi komputasi, di mana hanya terdapat 1 dimensi dalam *greyscale*, sedangkan pada *color space* lain yang melibatkan warna seperti RGB (*Red Green Blue*)

memiliki 3 dimensi. Selain itu, metode analisa yang digunakan tidak membandingkan komposisi warna, melainkan dari pola citra uang. Meskipun analisa dalam *color space greyscale* beresiko mengurangi akurasi dari program rekognisi, namun metode yang digunakan pada tugas akhir ini sudah memiliki akurasi yang cukup memuaskan. Program MATLAB untuk melakukan fungsi tersebut adalah sebagai berikut:

```
Q=(imread(img_crop));  
Q2=rgb2gray(Q);
```

Selanjutnya dilakukan *resizing* pada citra uang yang berfungsi untuk mengecilkan ukuran citra uang. Dengan mengecilkan ukuran piksel citra uang, proses analisa citra dapat berlangsung dengan lebih ringan dan cepat. Citra digital uang dikecilkan menjadi 173x80 piksel. Ukuran 80 piksel dianggap sudah mampu mempertahankan karakteristik setiap citra namun cukup ringan untuk dikomputasi. Lebar piksel yang diperoleh merupakan hasil penyesuaian lebar citra terhadap tinggi yang telah ditetapkan untuk mempertahankan proporsi (*aspect ratio*) citra. Program untuk melakukan *resizing* adalah sebagai berikut:

```
Q3=double(Q2);  
Q3=imresize(Q3, [80,NaN]);
```

3.2.3. Analisa Citra Digital Uang

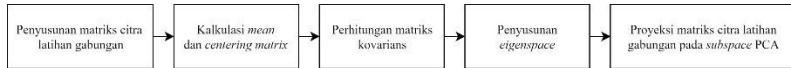
Analisa citra digital uang pada tugas akhir ini dilakukan menggunakan 2 metode, yaitu PCA dan LDA. Karena masing-masing metode memiliki kelebihan dan kekurangannya masing-masing, kedua metode analisa akan diuji terlebih dahulu untuk menentukan metode yang lebih cocok dalam mendeteksi nominal citra uang. Maka dari itu, analisa citra digital uang dibagi menjadi 3, yaitu PCA (*Principal Component Analysis*), LDA (*Linear Discriminant Analysis*), dan kalkulasi *euclidean distance*.

3.2.3.1. *Principal Component Analysis* (PCA)

Analisa metode PCA dilakukan dengan membentuk suatu matriks baru dengan dimensi yang rendah namun tetap mempertahankan variasi matriks aslinya. Dalam tugas akhir ini, PCA digunakan untuk membentuk suatu matrik transformasi yang disebut *eigenspace* berdasarkan matriks citra latihan yang merupakan kumpulan citra-citra

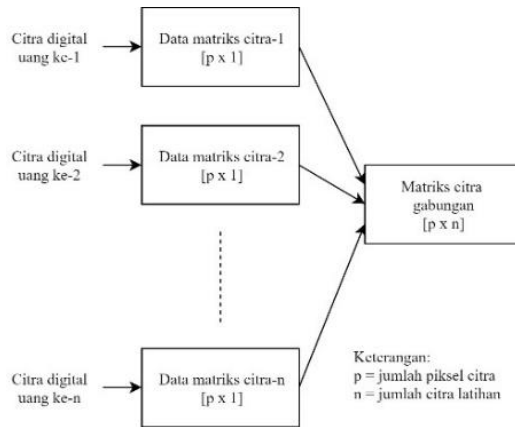
uang untuk setiap nominal uang yang digunakan. Secara garis besar, program analisa citra uang dengan menggunakan metode PCA dibagi menjadi 2, penyusunan *database* citra latihan dalam *subspace* PCA dan proyeksi citra pengujian dalam *subspace* PCA.

Diagram blok dari penyusunan matriks *database* citra latihan dari PCA ditunjukkan pada gambar 3.21.



Gambar 3.21 Diagram blok penyusunan matriks *database* citra latihan dalam *subspace* PCA

Pertama-tama dilakukan penyusunan matriks citra gabungan dari setiap citra latihan. Citra latihan terdiri dari sampel-sampel citra uang untuk masing-masing nominal uang dengan total citra latihan sebanyak n citra. Masing-masing citra uang berukuran p piksel disusun menjadi suatu matriks 1 kolom yang terdiri dari data setiap piksel citra dengan ukuran $p \times 1$. Setiap anggota matriks merupakan suatu angka yang merepresentasikan tingkat kecerahan (*brightness*) dari piksel dan citra latihan tertentu. Kemudian, semua citra disusun menjadi satu matriks gabungan yang merupakan matriks seluruh citra latihan dengan ukuran $p \times n$. Pada tugas akhir ini, citra latihan uang memiliki sampel sebanyak 112 citra yang terdiri dari 16 jenis citra uang dengan masing-masing 7 sampel citra dan dengan ukuran total piksel per gambar sebesar 13840 piksel. Matriks citra latihan gabungan ini merupakan masukan dari PCA, di mana akan diperoleh *eigenvectors* atau sumbu baru (*Principal Component*) yang memaksimalkan variasi data. Dengan kata lain, *input* dari metode PCA ini adalah matriks yang merupakan kumpulan tingkat kecerahan (*brightness*) untuk setiap piksel citra-citra latihan. Penyusunan matriks citra latihan gabungan ini diilustrasikan pada diagram blok gambar 3.22, di mana citra-citra latihan disusun berurutan dari nominal Rp 10.000,00 orientasi 1 hingga nominal Rp 100.000,00 orientasi 4.



Gambar 3.22 Penyusunan matriks citra uang gabungan

Program untuk melakukan fungsi tersebut adalah sebagai berikut:

```

Database_Training = [];
for i=1:112
    s = sprintf('Training/%i.jpg',i);
    X=(imread(s));
    Database_Training = [Database_Training, X(:)];
End

```

Data citra dapat langsung dianalisa dengan memasukkan variabel matriks citra gabungan (X) dan banyaknya *eigenvector* yang diinginkan untuk dijadikan *eigenspace* (N) pada program. Banyaknya *principal component* (*eigenvector*) yang dapat diambil memiliki jumlah maksimal $n-1$ dengan n adalah angka terkecil antara jumlah baris dan kolom matriks. Batas maksimal $n-1$ diperoleh akibat pengurangan dengan *mean* pada matriks data citra, sehingga ada satu vektor nol yang bukan *eigenvector*. Semakin banyak *eigenvector* yang dicari, maka variasi data yang dipertahankan semakin baik. Pada tugas akhir ini, jumlah *eigenvector* yang diambil adalah 111 yang merupakan batas maksimal jumlah *eigenvector* yang dapat diambil. Diambil jumlah *eigenvector* maksimal karena pengurangan dimensi dari 13840 menjadi 111 dianggap sudah cukup signifikan, sedangkan diharapkan program mampu memiliki akurasi setinggi mungkin dengan mengambil batas maksimal *eigenvector* yang dipertahankan. Program pada tahapan ini adalah sebagai berikut:

```

X = Database_Training;
N = 111;
modelPCA = perform_pca_PhD(X, N);

```

Program ini akan memberikan 4 variabel *output*, yaitu “modelPCA.P” yang merupakan *mean* dari matriks citra gabungan, “modelPCA.dim” yang merupakan dimensionalitas dari *subspace* PCA, “modelPCA.W” yang merupakan matriks transformasi *eigenspace*, dan “modelPCA.train” yang merupakan proyeksi matriks citra latihan gabungan dalam *subspace* PCA.

Pada program ini, pertama-tama dilakukan penyusunan *centering matrix*, yaitu matriks yang sudah dikurangi rata-rata dari matriks gabungan. *Centering matrix* dibutuhkan untuk memperoleh matriks kovarians pada langkah berikutnya. Proses ini dijelaskan pada persamaan berikut:

$$Mean = \frac{1}{n} \sum_{i=1}^n X_i \quad (3.1)$$

$$X^{-i} = X^i - Mean \quad (3.2)$$

$$\bar{X} = [X^{-1} X^{-2} \dots X^{-n}] \quad (3.3)$$

Dengan n adalah jumlah citra latihan, dan i adalah indikasi citra ke- i . Program untuk memperoleh *centering matrix* tersebut adalah sebagai berikut:

```

[a,b]=size(X);
modelPCA.P = mean(X,2);
X = X-repmat(modelPCA.P,1,b);

```

Fungsi *mean(X,2)* mengalkulasi rata-rata setiap baris X dan disusun dalam satu vektor kolom. Fungsi *repmat* melakukan repetisi dari vektor rata-rata untuk mengurangi setiap kolom dalam matriks citra gabungan.

Setelah itu dilanjutkan dengan mencari *eigenvector* dari matriks kovarians. Matriks kovarians dapat diperoleh dengan mengalikan *centering matrix* dengan *centering matrix transpose* yang ditunjukkan pada persamaan berikut:

$$C = \bar{X} \bar{X}^T \quad (3.4)$$

Program untuk melakukan fungsi tersebut adalah sebagai berikut:

```
[U,V,L] = svd(X*X');
Clear L;
Clear U;
```

Pada program ini, *eigenvector* dicari dengan menggunakan fungsi MATLAB “*Singular Value Decomposition*”, dimana fungsi ini akan memberikan *output* berupa variabel U , V , dan L . Variabel U mengandung *eigenvectors* dari XX^T , variabel V mengandung *eigenvectors* dari X^TX , dan S mengandung akar kuadrat dari *eigenvalues* dari XX^T dan X^TX . Pada program ini hanya dibutuhkan *eigenvectors* dari XX^T , maka dari itu hanya digunakan variabel U , sedangkan variabel lainnya dihapus untuk mengurangi penggunaan memori. *Eigenvector-eigenvector* yang diambil kemudian disusun menjadi *eigenspace*, yaitu matriks transformasi yang dapat digunakan untuk mengubah matriks *database* asli berdimensi tinggi menjadi matriks dengan dimensi rendah namun tetap mempertahankan variasinya. Program untuk menentukan *eigenspace* adalah sebagai berikut:

```
modelPCA.W = normc(U(:,1:N));
```

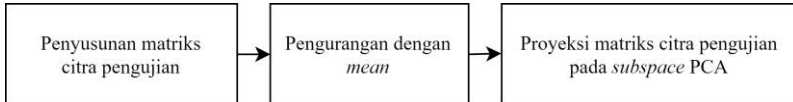
Fungsi $U(:,1:N)$ melakukan ekstraksi data matriks *eigenvectors* dengan mengambil seluruh baris matriks dari kolom 1 sampai kolom ke- N . Pada program di atas juga dilakukan normalisasi agar semua data ditransformasi dalam distribusi normal. Variabel “modelPCA.W” menyimpan matriks transformasi atau *eigenspace* yang memiliki ukuran $p \times N$. Pada tugas akhir ini, “modelPCA.W” akan memiliki ukuran 13840 x 111.

Langkah berikutnya adalah memproyeksikan matriks citra latihan gabungan dalam *subspace* PCA. Langkah ini dilakukan dengan mengalikan matriks transformasi *eigenspace* yang telah di-*transpose* menjadi berukuran $N \times p$ dengan matriks citra latihan gabungan asli yang telah dikurangi *mean* (*centering matrix*) berukuran $p \times n$ ke dalam *eigenspace*. Hasil akhir dari proyeksi ini menghasilkan matriks *database* citra latihan dengan ukuran $N \times n$ dan disimpan dalam variabel “modelPCA.train”. Kemudian, seluruh variabel matriks “modelPCA.mat” disimpan agar dapat digunakan dalam deteksi citra pengujian. Pada tugas akhir ini, matriks “modelPCA.train” memiliki ukuran 111 x 112. Program untuk melakukan proyeksi ini adalah sebagai berikut:

```
modelPCA.train = modelPCA.W'*X;
```

```
save('modelPCA.mat', 'modelPCA');
```

Setelah memperoleh matriks *database* citra latihan, citra-citra digital uang yang akan diuji juga akan diproyeksikan ke dalam *subspace* PCA. Blok diagram untuk memproyeksikan citra pengujian ditunjukkan pada gambar 3.23.



Gambar 3.23 Diagram blok penyusunan matriks pengujian dalam *subspace* PCA

Proyeksi citra pengujian ke dalam *subspace* PCA menggunakan variabel-variabel yang telah disimpan pada penyusunan matriks *database*, yakni *mean* (modelPCA.P) dan *eigenspace* (modelPCA.W). Citra pengujian disusun ke dalam matriks 1 kolom seperti pada matriks citra latihan. Kemudian, matriks citra dikurangi oleh *mean* yang telah diperoleh sebelumnya. Matriks diproyeksikan dengan mengalikan matriks *transpose eigenspace* dengan matriks citra pengujian. Program untuk melakukan langkah-langkah di atas adalah sebagai berikut:

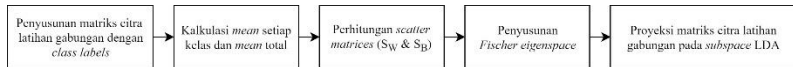
```
load('modelPCA.mat');
m = sprintf('Testing/1.jpg');
Y=(imread(m));
Matrix_Testing = [];
Matrix_Testing = [Matrix_Testing, Y(:)];
T=[];
T(:,1) = Matrix_Testing(:,1) - modelPCA.P;
Y = modelPCA.W'*T;
```

3.2.3.2. Linear Discriminant Analysis (LDA)

Metode analisa LDA merupakan metode klasifikasi *supervised* di mana data-data citra setiap nominal uang dikelompokkan dalam kelas-kelas dengan diberikan label. LDA akan melakukan transformasi matriks ke arah di mana separasi antar kelas memiliki nilai semaksimal mungkin.

Seperti proses pada PCA, pertama-tama akan dibentuk matriks citra gabungan sebagai matriks *database* citra latihan, namun pada LDA akan diberikan *class label* untuk masing-masing kelas citra nominal uang. Karena terdapat 16 jenis citra uang pada tugas akhir ini, maka terdapat

16 pembagian kelas pada program LDA. Setelah itu akan diperoleh *eigenspace* Fischer yang merupakan matriks transformasi dengan daya diskriminan yang tinggi, di mana matriks citra latihan dan pengujian akan diproyeksikan. Diagram blok dari penyusunan matriks *database* citra latihan dengan LDA ditunjukkan pada gambar 3.24.



Gambar 3.24 Diagram blok penyusunan matriks *database* citra latihan dalam *subspace* LDA

Program ini akan melakukan LDA pada masukan matriks citra latihan gabungan (X), matriks 1×16 yang merupakan *class label* dari citra-citra latihan dengan 16 kelas (*ids*), dan jumlah vektor basis yang dipertahankan dalam proses LDA (N) dengan maksimal jumlah kelas dikurang 1. Pada tugas akhir ini, jumlah vektor basis yang dipertahankan adalah 15 agar memperoleh separasi antar kelas semaksimal mungkin. Selain itu pada fungsi ini juga dilakukan PCA terlebih dahulu untuk mengurangi dimensi dari matriks citra. Program akan memberikan 4 variabel *output*, yaitu “modelLDA.P” yang merupakan *mean* dari matriks citra latihan gabungan, “modelLDA.dim” adalah dimensionalitas LDA, “modelLDA.W” yang merupakan matriks transformasi LDA untuk memaksimalkan separasi, dan “modelLDA.train” yang merupakan *database* citra latihan dalam *subspace* LDA. Program keseluruhan LDA adalah sebagai berikut:

```

Database_Training = [];
for i=1:112
    s = sprintf('Training/%i.jpg',i);
    X=(imread(s));
    Database_Training = [Database_Training, X(:)];
End

X = Database_Training;
ids = [1 1 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 3 4 4 4 4
4 4 4 5 5 5 5 5 5 5 6 6 6 6 6 6 7 7 7 7 7 7 8 8 8 8
8 8 9 9 9 9 9 9 9 10 10 10 10 10 10 11 11 11 11 11 11
11 12 12 12 12 12 12 12 13 13 13 13 13 13 13 14 14 14 14
14 14 14 15 15 15 15 15 15 15 16 16 16 16 16 16 16];
N = 15;

modelLDA = perform_lda_PhD(X, ids, N);
  
```

Pada program pertama-tama akan dicari *eigenspace* PCA untuk mengurangi dimensi dari data. Langkah-langkah untuk mencari *eigenspace* PCA sama seperti metode PCA yang telah dijelaskan sebelumnya, yakni meliputi *centering matrix*, pencarian *eigenvector*, dan penyusunan *eigenspace*. Selain itu, juga dibentuk suatu matriks *whitening transformation*, yaitu matriks transformasi linear yang melakukan transformasi pada sebuah vektor dengan matriks kovarians yang diketahui menjadi satu set variabel baru dengan kovarians berupa matriks identitas. Program dalam fungsi untuk mencari *eigenspace* tersebut adalah sebagai berikut:

```
modelLDA.P = mean(X,2);
modelLDA.dim = n;
Fi = X-repmat(modelLDA.P,1,b);
[U,V,L] = svd(Fi'*Fi);
clear V L
PCA = normc(Fi*U(:,1:b-1));
clear U

tmp = sqrt(diag(PCA'*Fi'*Fi'*PCA));
PCA = PCA./repmat(tmp',a,1);
```

Setelah itu dilakukan komputasi *mean* untuk setiap kelas. Komputasi ini dilakukan dengan program sebagai berikut:

```
[a,b]=size(X);
id_unique = unique(ids);
[c,d]=size(id_unique);
num_of_class = max([c,d]);

meanclass = zeros(a,num_of_class);
class_id = zeros(1,num_of_class);
for i=1:num_of_class
    [dummy,ind]=find(id_unique(i)==ids);
    meanclass(:,i) = mean(X(:,ind),2);
end
```

Lalu dilanjutkan dengan mencari *within class scatter matrix* (S_W), yaitu matriks yang menghitung tingkat penyebaran antar citra dalam satu kelas; dan *between class scatter matrix* (S_B), yaitu matriks yang menghitung tingkat penyebaran antar kelas. Persamaan kedua matriks ini adalah sebagai berikut:

$$S_i = \sum_{x \in X_i} (x - m_i)(x - m_i)^T \quad (3.5)$$

$$S_W = \sum_{i=1}^C S_i \quad (3.6)$$

$$S_B = \sum_{i=1}^C n_i(m_i - m)(m_i - m)^T \quad (3.7)$$

Program untuk mencari S_W dan S_B adalah sebagai berikut:

```

Fiw = zeros(a,b);
cont = 1;
for i=1:num_of_class
    [dummy,ind]=find(id_unique(i)==ids);
    num_of_sampl = length(ind);
    for j=1:num_of_sampl
        Fiw(:,cont) = X(:,ind(j))-meanclass(:,i);
        cont=cont+1;
    end
end

Fib = meanclass-repmat(modelLDA.P,1,num_of_class);

Sb = PCA(:,1:max_dim)'*Fib*Fib'*PCA(:,1:max_dim);
Sw = PCA(:,1:max_dim)'*Fiw*Fiw'*PCA(:,1:max_dim);

```

Selanjutnya fungsi dilanjutkan dengan mencari *generalized eigenvector*, yaitu *eigenvector* yang menjawab persamaan berikut:

$$S_B v = \lambda S_W v \quad (3.8)$$

dengan v adalah *generalized eigenvectors* dan λ adalah *generalized eigenvalues* dari kedua matriks S_B dan S_W . Sama seperti pada PCA, digunakan fungsi SVD untuk mencari *eigenvector*. *Generalized eigenvectors* yang diperoleh kemudian akan disusun menjadi *eigenspace* Fischer dengan jumlah *eigenvectors* yang diambil sebanyak N . Pada tugas akhir ini diambil nilai N maksimal yaitu 15 untuk memperoleh separasi maksimal. Setelah itu, matriks citra latihan gabungan diproyeksikan ke dalam *subspace* LDA dengan daya diskriminan tinggi yang menghasilkan *database* citra latihan. Program untuk melakukan hal

ini adalah sebagai berikut:

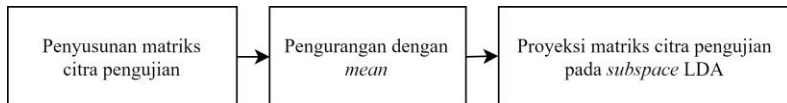
```
[U1,D1,QP] = svd(Sww);
U1_scaled = normc(U1);
clear U1_norm U1 D1 QP

Sbbb = U1_scaled'*Sbb*U1_scaled;
[U1,D1,QP] = svd(Sbbb);

fisher = PCA(:,1:max_dim)*U1_scaled*U1(:,1:num_of_class-1);
clear PCA U1 U1_scaled

modelLDA.W = normc(fisher(:,1:num_of_class-1));
modelLDA.train = modelLDA.W'*Fi;
```

Seperti pada PCA, citra uang untuk pengujian juga harus diproyeksikan pada *subspace* LDA. Proyeksi dilakukan dengan mengurangi matriks citra pengujian dengan *mean* yang telah diperoleh sebelumnya dan mengalikan *eigenspace* Fischer yang telah dilakukan *transpose* dengan matriks citra pengujian. Blok diagram proses ini ditunjukkan pada gambar 3.25.



Gambar 3.25 Diagram blok penyusunan matriks pengujian dalam *subspace* LDA

Program untuk menyusun matriks pengujian dalam *subspace* LDA adalah sebagai berikut:

```
load('modelLDA.mat');
m = sprintf('Testing/1.jpg');
Y=(imread(m));
Matrix_Testing = [];
Matrix_Testing = [Matrix_Testing, Y(:)];
T=[];
T(:,1) = Matrix_Testing(:,1) - modelLDA.P;
Y = modelLDA.W'*T;
```

3.2.3.3. Kalkulasi *Euclidean Distance*

Setelah diperoleh matriks *database* citra latihan, baik menggunakan PCA maupun LDA, dilakukan kalkulasi jarak Euclidean (*Euclidean distance*) untuk mendeteksi citra uang pengujian terhadap *database* citra latihan. Kalkulasi jarak Euclidean ditunjukkan dalam persamaan berikut:

$$d(X, Y) = d(Y, X) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (3.8)$$

Pada tugas akhir ini, jarak Euclidean akan dikalkulasi antara matriks citra pengujian dengan seluruh sampel citra latihan. Karena sampel *database* citra latihan pada tugas akhir ini terdapat 112 sampel gambar, maka kalkulasi jarak Euclidean dilakukan sebanyak 112 kali untuk satu citra pengujian. Program untuk mengalkulasi jarak Euclidean yang digunakan pada tugas akhir ini adalah sebagai berikut:

```
distance = [];
for itrain=1:112
    distance((itrain),(1)) =
    sqrt(sum(((modelLDA.train(:,(itrain))) -
    (Y(:,(1))))).^2));
end
```

Pada program di atas, metode yang digunakan adalah LDA. Program di atas juga berlaku untuk PCA jika matriks yang dipanggil adalah matriks latihan PCA dan matriks citra pengujian diproyeksikan dalam *subspace* PCA.

Selanjutnya dilakukan klasifikasi citra pengujian dengan mencari jarak Euclidean terkecil. Posisi baris dari jarak Euclidean terkecil menentukan penentuan jenis klasifikasi nominal citra uang yang terdeteksi. Hal ini dijelaskan pada tabel 3.3 berikut.

Tabel 3.3 Rekognisi berdasarkan posisi baris jarak terkecil

Posisi baris d_{\min} ($d_{\min} < threshold$)	Nominal Uang terdeteksi
1-28	Rp 10.000,00
29-56	Rp 20.000,00
57-84	Rp 50.000,00
85-112	Rp 100.000,00

Pada tugas akhir ini juga diberikan *distance threshold* di mana ketika nilai jarak Euclidean terkecil melewati *distance threshold*, maka citra pengujian dianggap gagal atau bukan merupakan citra yang terdapat di dalam *database* latihan. Hal ini dilakukan untuk mengantisipasi terdeteksinya uang palsu, citra bukan uang, maupun uang-uang yang tidak diterima oleh *vending machine*. Hasil deteksi berdasarkan jarak Euclidean kemudian diteruskan ke Arduino Mega melalui komunikasi serial dengan kode sesuai dengan hasil deteksi. Program untuk melakukan hal ini adalah sebagai berikut:

```
I = find(distance==min(distance(:)));
NM = min(distance);

if (NM > 457)
    fprintf(s, 100);
else
    if (I <= 28)
        hasil = 'uang 10.000';
        fprintf(s, 101);
    elseif (I >= 29) && (I <= 56)
        hasil = 'uang 20.000';
        fprintf(s, 102);
    elseif (I >= 57) && (I <= 84)
        hasil = 'uang 50.000';
        fprintf(s, 103);
    elseif (I >= 85) && (I <= 112)
        hasil = 'uang 100.000';
        fprintf(s, 104);
    end
end
```

3.2.4. Sistem Pembayaran Mesin Penjual Swalayan

Sistem pembayaran tunai mesin penjual swalayan dilakukan oleh perangkat keras Arduino Mega 2560 dengan perangkat lunak Arduino IDE dan oleh perangkat lunak MATLAB pada PC. Program pada Arduino Mega 2560 meliputi seluruh perangkat keras sistem, sedangkan program pada MATLAB meliputi komunikasi serial dengan Arduino untuk menerima perintah pengambilan gambar dan menginformasikan hasil deteksi gambar.

Sistem pembayaran *vending machine* yang diprogram pada Arduino menggunakan *library* <Wire.h> dan <LiquidCrystal_I2C.h> untuk pemrograman LCD dengan I2C, dan *library* <Keypad.h> untuk

pemrograman *keypad* 4x4. Program sistem dimulai dari inisialisasi *library* yang digunakan dan inisialisasi variabel-variabel dan pin yang digunakan dalam program. Pada program juga dilakukan inisialisasi variabel “saldo” yang menyimpan saldo sisa kembalian dan variabel “uang” yang menyimpan nilai nominal citra uang yang terdeteksi. Kedua variabel saldo dan uang diinisialisasi dengan nilai 0. Program untuk melakukan hal ini adalah sebagai berikut:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
#include <Keypad.h>

int uang = 0;
int saldo = 0;
```

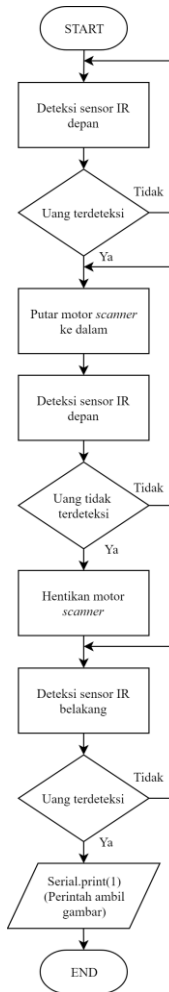
Selanjutnya dilanjutkan program *setup* pada Arduino untuk melakukan *setup* serial, *setup* pin perangkat elektronik, *setup* kondisi awal modul relay, *setup* kondisi LED, dan menampilkan tampilan awal LCD pada sistem. Program untuk melakukan *setup* kondisi awal dari relay dan LED adalah sebagai berikut:

```
digitalWrite(pinOut, HIGH); // pin relay 1
digitalWrite(pinOut_2, HIGH); // pin relay 2
digitalWrite(pinLED_1, LOW); // LED1
digitalWrite(pinLED_2, LOW); // LED2
digitalWrite(pinLED_3, LOW); // LED3
digitalWrite(pinLED_4, LOW); // LED4
```

Pin dari relay diberikan *high* karena modul relay yang digunakan bersifat *active low* di mana *input* akan aktif ketika diberikan kondisi *low*, sedangkan rangkaian LED bersifat *active high* di mana LED akan menyala ketika diberikan masukan *high*. *Setup* dari kedua perangkat di atas dilakukan untuk melakukan *reset* ketika sistem baru mulai dijalankan.

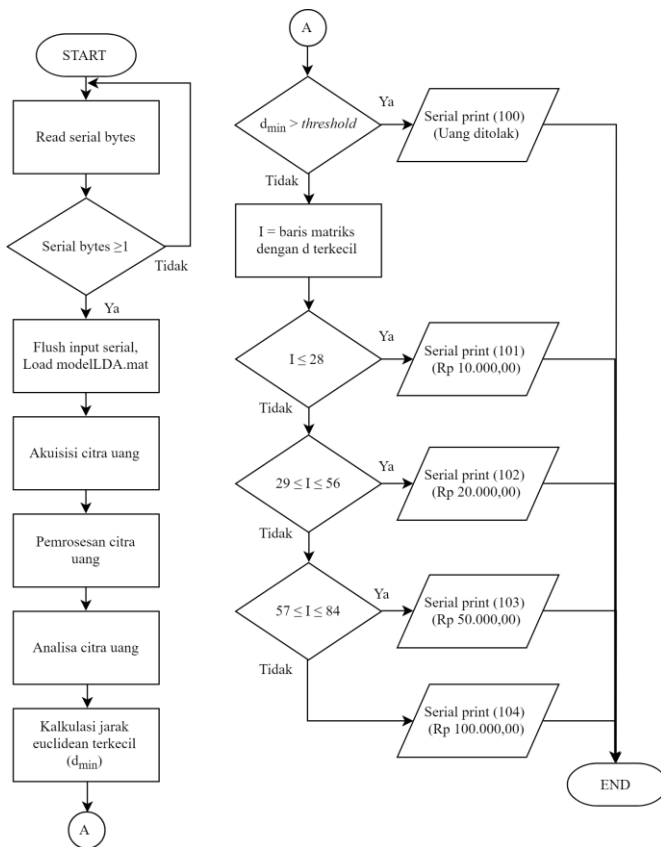
Kemudian sistem dilanjutkan dengan deteksi sensor IR pada *scanner* uang. Seperti yang sudah dijelaskan sebelumnya bahwa terdapat dua sensor IR pada *scanner* uang yang dibagi menjadi sensor IR depan dan belakang. Sensor IR bagian depan bertugas untuk memutar motor *scanner* uang selama uang masih terdeteksi oleh sensor, sedangkan sensor IR bagian belakang bertugas untuk mengirimkan perintah serial kepada PC untuk melakukan pengambilan citra uang oleh kamera.

Penggunaan dua sensor IR ini dilakukan untuk mencegah pengambilan citra oleh kamera ketika uang tidak jadi dimasukkan oleh pengguna mesin penjual swalayan. *Flowchart* dari program *scanner* uang pada Arduino Mega 2560 ditunjukkan pada gambar 3.26.



Gambar 3.26 *Flowchart* dari *scanner* uang pada sistem

Pada tahap ini, Arduino Mega 2560 menunggu hasil analisa citra uang pada perangkat lunak MATLAB yang dilakukan pada PC. Ketika MATLAB menerima perintah serial, MATLAB melakukan akuisisi citra digital uang, pemrosesan citra digital uang, analisa citra digital uang, dan kalkulasi jarak Euclidean seperti yang sudah dijelaskan pada subbab sebelumnya. Hasil kalkulasi jarak Euclidean merupakan hasil pengenalan citra uang, yang kemudian diteruskan ke Arduino Mega 2560 melalui komunikasi serial. *Flowchart* untuk melakukan proses ini pada MATLAB ditunjukkan pada gambar 3.27.



Gambar 3.27 *Flowchart* pengenalan citra uang pada MATLAB

Hasil deteksi citra uang diteruskan ke Arduino Mega 2560 untuk mengatur variabel “uang” sesuai nominal uang yang terdeteksi. Kemudian, variabel uang ini ditambahkan ke dalam variabel “saldo”. Ketika saldo yang tersimpan bukan nol, sistem akan menampilkan jumlah saldo yang tersimpan pada LCD. Sedangkan ketika saldo adalah nol, maka LCD akan menampilkan tampilan awal *vending machine*. Saldo ini kemudian dapat digunakan untuk melakukan transaksi dengan *keypad*. Perintah-perintah pada *keypad* dijelaskan pada tabel 3.4.

Tabel 3.4 Perintah masukan pada *keypad*

Keypad	Perintah
A	Batalkan transaksi
B	Lakukan pengembalian uang
1	Beli produk 1 (Rp 10.000,00)
2	Beli produk 2 (Rp 20.000,00)
3	Beli produk 3 (Rp 50.000,00)
4	Beli produk 4 (Rp 80.000,00)

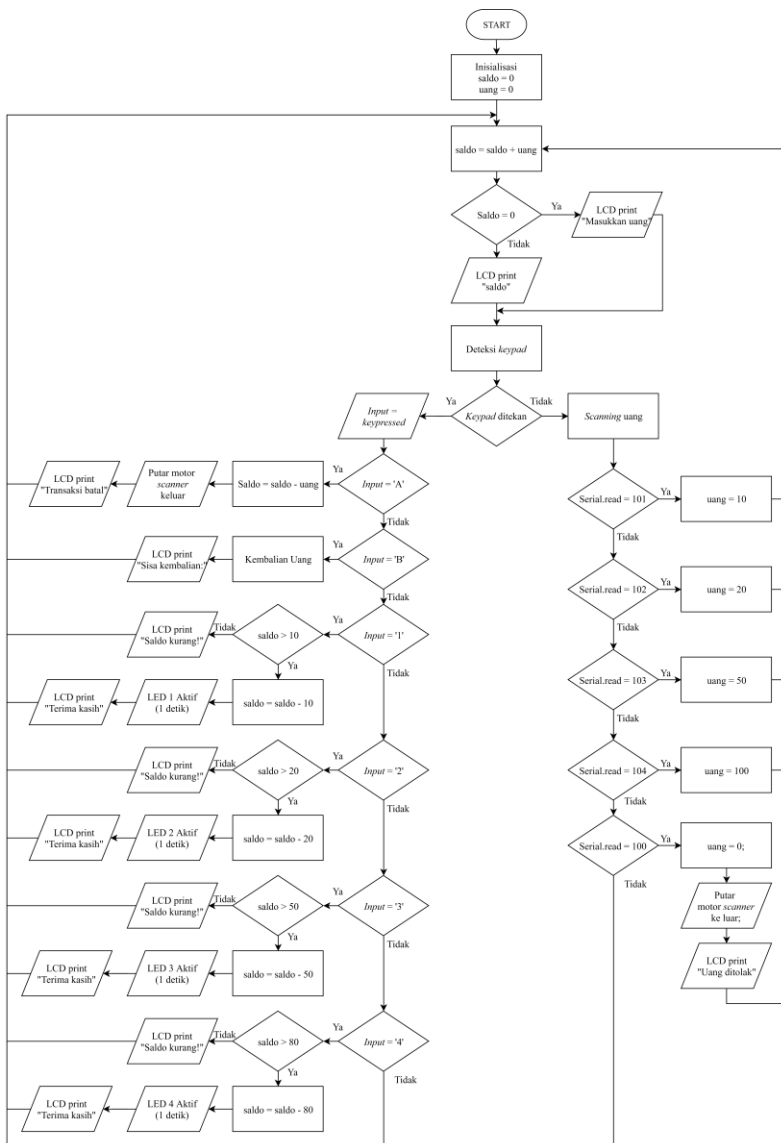
Perlu diingat bahwa pembatalan transaksi hanya bisa dilakukan pada uang terakhir yang dimasukkan dan sebelum transaksi dilakukan, ketika dilakukan sebaliknya maka sistem akan menolak perintah. Selain itu, pengembalian uang hanya bisa dilakukan ketika transaksi sudah dilakukan.

Pengembalian uang dilakukan ketika perintah ‘B’ pada *keypad* ditekan oleh pengguna. Program pengembalian uang akan mengembalikan uang sesuai saldo yang tersisa setelah dilakukan transaksi. Apabila saldo yang tersisa lebih dari 50, maka pengembalian uang bagian atas (Rp 50.000,00) akan memberikan pengembalian sebanyak 1 lembar dengan memutar motor *stepper* sebanyak 360° dan motor DC 5V ke luar dan mengurangi saldo dengan 50. Kemudian, program akan kembali melakukan pengecekan terhadap sisa saldo sampai saldo bernilai kurang dari 50. Selanjutnya dilakukan pengecekan sisa saldo untuk pengembalian Rp 10.000,00. Ketika jumlah saldo tersisa lebih besar dari 10 dan lebih kecil dari 50, program akan memberikan pengembalian Rp 10.000,00 sebanyak 1 lembar. Sama seperti proses sebelumnya, saldo akan dicek kembali hingga saldo tersisa bernilai 0. Untuk lebih jelasnya, program dari pengembalian uang dijelaskan pada *flowchart* gambar 3.28.



Gambar 3.28 *Flowchart* program pengembalian uang

Flowchart dari keseluruhan sistem transaksi mesin penjual swalayan ditunjukkan pada gambar 3.29.



Gambar 3.29 Flowchart sistem transaksi mesin penjual swalayan

BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang pada bab sebelumnya. Bab ini bertujuan untuk mendapatkan data pengujian bagi masing-masing bagian sistem, yaitu meliputi pengujian *scanner* uang, pengujian pengembalian uang, pengujian akuisisi citra uang, pengujian akuisisi citra uang, pengujian pemrosesan citra uang, pengujian pengenalan citra uang, pengujian dengan uang palsu, dan pengujian keseluruhan sistem. Selain itu juga akan dijelaskan mengenai realisasi sistem dan inisialisasi sensor inframerah.

4.1. Realisasi Sistem

Pada subbab ini akan ditunjukkan realisasi mekanik dan elektronik sistem secara keseluruhan. Realisasi mekanik dan elektronik bagian dalam dan tampak luar realisasi sistem secara keseluruhan ditampilkan pada gambar 4.1. Sistem yang ditampilkan tidak termasuk PC yang dihubungkan secara serial, karena PC diletakkan di luar sistem.



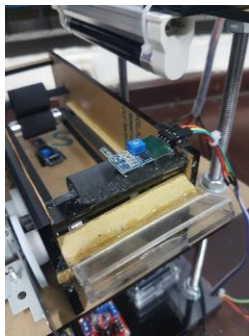
Gambar 4.1 (a) Realisasi mekanik dan elektronik sistem (b) Realisasi sistem secara keseluruhan

Tabel 4.1. Spesifikasi sistem

Dimensi	(35 x 45 x 85) cm
Mikrokontroler	Arduino Mega 2560
Sensor	2 Sensor IR TCRT5000
Aktuator	1 Motor DC 12V, 2 Motor DC 5V, 2 Motor <i>stepper</i> 12V
Uang yang diterima	Uang kertas rupiah tahun emisi 2016 pecahan: <ul style="list-style-type: none">• Rp 10.000,00• Rp 20.000,00• Rp 50.000,00• Rp 100.000,00
Uang kembalian yang tersedia	Uang kertas rupiah pecahan: <ul style="list-style-type: none">• Rp 10.000,00• Rp 50.000,00

4.2. Inisialisasi Sensor IR TCRT5000

Sensor IR TCRT5000 digunakan dalam deteksi keberadaan uang pada *scanner* uang. Terdapat 2 sensor IR yang digunakan, yakni sensor IR depan dan sensor IR belakang. Sensor IR TCRT5000 memiliki jarak deteksi dari 0.2 sampai 15 mm. Inisialisasi sensor IR perlu dilakukan untuk kedua sensor yang digunakan agar sensitivitas dari sensor sesuai dengan keberadaan uang. Posisi dari sensor IR depan pada *scanner* uang ditunjukkan pada gambar 4.2, dan posisi dari sensor IR belakang pada *scanner* uang ditunjukkan pada gambar 4.3.



Gambar 4.2 Letak sensor IR depan



Gambar 4.3 Letak sensor IR belakang

Inisialisasi dilakukan dengan mengukur *output* analog yang terukur pada kondisi terdapat uang maupun saat kondisi tidak ada uang. Pengukuran besaran analog dan digital dilakukan dengan mengamati serial monitor Arduino yang menampilkan *output* analog dan digital sensor dengan *baud rate* 9600 bps. Dari besaran analog ini, sensitivitas sensor diatur secara manual agar sensor dapat memberikan *output* digital yang sesuai ketika diberikan uang. Hasil inisialisasi sensor IR untuk depan dan belakang dapat dilihat pada tabel 4.2.

Tabel 4.2. Hasil inisialisasi sensor IR depan dan belakang

Sensor IR	Kondisi tidak ada uang			Kondisi ada uang		
	Analog		Digital	Analog		Digital
	Min	Max		Min	Max	
Depan	958	960	1	561	680	0
Belakang	1010	1012	1	725	727	0

Dari tabel di atas dapat disimpulkan bahwa *output* digital sudah sesuai seperti dengan yang diinginkan, yaitu perubahan kondisi biner dari pin digital ketika terdapat uang di hadapan sensor IR. Kondisi dari pin analog untuk sensor IR depan ketika ada uang lebih beragam karena keadaan uang saat masuk ke dalam mulut *scanner* beragam, tidak seperti pada sensor IR belakang di mana keadaan uang sudah tetap.

4.3. Pengujian Scanner Uang

Pengujian *scanner* uang dilakukan dengan memasukkan uang dengan kondisi acak sebanyak 30 kali dan melakukan pengeluaran uang sebanyak 30 kali. Pengujian ini dilakukan untuk menguji kinerja dari mekanik dan elektronik *scanner* uang dalam menarik uang ke dalam maupun mendorong uang kembali ke luar. Hasil pengujian ditunjukkan pada tabel 4.3.

Tabel 4.3. Hasil pengujian *scanner* uang

Pengujian	Berhasil	Gagal	Error
Penarikan uang	30	0	0%
Pendorongan uang	23	7	23,3%

Dari hasil pengujian, diperoleh bahwa penarikan uang sudah bekerja dengan baik dengan *error* sebesar 0%, sedangkan masih terdapat *error* 23,3% pada pendorongan uang. *Error* pada pengujian berupa tersangkutnya uang yang dikeluarkan pada mulut mekanik *scanner* uang sehingga uang harus dikeluarkan secara manual. Dari pengamatan selama pengujian, sebagian besar kejadian tersangkutnya uang terjadi ketika uang yang digunakan lusuh atau dalam kondisi tidak lurus.

4.4. Pengujian Pengembalian Uang

Pengujian pengembalian uang dilakukan dengan melakukan pengeluaran kembalian uang untuk masing-masing nominal sebanyak 20 kali. Pengujian dilakukan 2 kali dengan mengeluarkan 10 lembar uang berturut-turut. Pengujian ini dilakukan untuk menguji kinerja mekanik dan elektronik pengembalian uang untuk nominal Rp 10.000,00 dan Rp 50.000,00. Hasil pengujian ditunjukkan pada tabel 4.4.

Tabel 4.4. Hasil pengujian pengembalian uang

Nominal	Berhasil	Gagal	Error
Rp 10.000,00	16	4	20%
Rp 50.000,00	17	3	15%

Dari hasil pengujian, dapat disimpulkan bahwa pengembalian uang masih memiliki *error* yang cukup besar, yaitu 20% untuk nominal Rp 10.000,00, dan 15% untuk nominal Rp 50.000,00. *Error* yang terjadi berupa keluarnya kembalian lebih dari 1 lembar untuk setiap putaran motor *stepper*. Dari hasil pengamatan selama pengujian, *error* cenderung

terjadi ketika masih terdapat 10 lembar uang dalam mekanik pengembalian, di mana kemungkinan terjadi karena tekanan uang antara mekanik dengan uang masih terlalu besar. Hal ini dapat diatasi dengan memberikan stok kembalian dengan jumlah lebih sedikit dari 10, namun hal ini juga menunjukkan bahwa mekanisme pengembalian uang masih kurang dapat diandalkan.

4.5. Pengujian Akuisisi Citra Uang

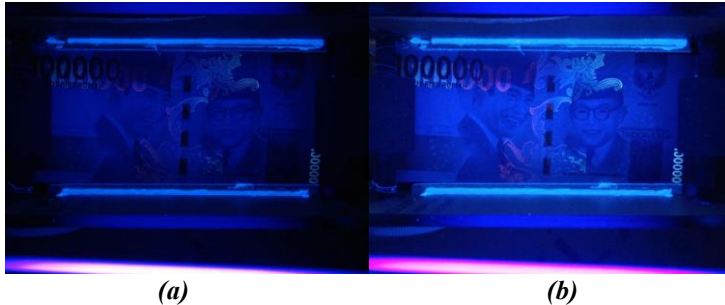
Pengujian akuisisi citra uang dilakukan dengan membandingkan gambar yang diambil secara manual dengan gambar yang diambil dengan menggunakan pengaturan kamera pada MATLAB yang digunakan. Pengaturan kamera diperoleh dengan proses *trial* dan *error* untuk memperoleh gambar yang diinginkan.

Proses *trial* dan *error* dilakukan secara manual terlebih dahulu pada perangkat lunak Logitech Gaming Software 9.00.42 yang mampu memberikan akses pengaturan kamera USB Logitech C922 dengan mudah. Hasil pengaturan yang diperoleh secara manual kemudian diterapkan di dalam program MATLAB. Dari proses *trial* dan *error* tersebut, diperoleh pengaturan kamera yang ditunjukkan pada tabel 4.5.

Tabel 4.5. Pengaturan kamera untuk akuisisi citra uang

Pengaturan	Value
<i>Focus</i>	15
<i>Exposure</i>	-4
<i>Backlight compensation</i>	1
<i>White balance</i>	4000
<i>Gain</i>	255

Dari hasil pengujian, dapat dilihat bahwa hasil citra yang diakuisisi dengan pengaturan kamera dalam program MATLAB sudah memberikan *output* citra yang cukup jelas dan sesuai dengan keinginan penulis. Maka dari itu dapat disimpulkan program akuisisi citra uang pada sistem sudah sesuai dengan yang diinginkan.



Gambar 4.4 (a) Citra akuisisi tanpa pengaturan. (b) Citra akuisisi dengan pengaturan

4.6. Pengujian Pemrosesan Citra Digital Uang





Pemrosesan citra digital uang terdiri dari *cropping*, perubahan *color space* menjadi *greyscale*, dan *resizing*. Pengujian dilakukan dengan mengambil citra uang untuk masing-masing proses.

Cropping dari citra yang diakuisisi dilakukan dengan mengambil piksel tertentu dari gambar. Pengambilan piksel ditentukan berdasarkan hasil *trial* dan *error* agar hasil pemotongan gambar dapat tepat mengambil citra uang saja dan membuang bagian yang tidak diperlukan. Ukuran *cropping* yang diperoleh adalah piksel dari koordinat awal $x=55$ ditarik sepanjang 520 piksel dan koordinat awal $y=90$ ditarik sepanjang 240 piksel.

Ukuran *resizing* ditentukan berdasarkan banyaknya piksel yang dianggap cukup untuk dianalisa. Ukuran *resizing* diambil 80 piksel untuk tinggi gambar atau dengan kata lain diambil 80 baris matriks data citra. Resolusi dari gambar dipertahankan sesuai resolusi awal. Hasil pemrosesan citra digital uang ditunjukkan pada tabel gambar 4.6.

Berdasarkan hasil pengujian, diperoleh bahwa seluruh program untuk setiap tahap dalam pemrosesan citra digital uang sudah sesuai dengan yang diinginkan. Program *cropping* mampu membuang bagian citra yang tidak dibutuhkan, program perubahan *color space* ke *greyscale* sudah mampu menghasilkan citra yang hanya terdiri dari kombinasi nilai hitam dan putih, dan program tahap *resizing* sudah memberikan ukuran akhir citra yang dianggap cukup untuk mengurangi beban komputasi namun tetap memperoleh akurasi yang memadai.

Tabel 4.6. Hasil pengujian pemrosesan citra digital uang










Citra asli dengan koordinat <i>cropping</i>		
<i>Cropping</i>		
Ubah ke <i>greyscale</i>		
<i>Resizing</i>		







4.7. Pengujian Pengenalan Citra Uang

Pengujian pengenalan citra uang meliputi analisa citra digital uang menggunakan metode PCA, analisa citra digital uang menggunakan metode LDA, dan kalkulasi jarak Euclidean. Pada bagian ini, pengujian dilakukan dengan menghitung jarak Euclidean dari 80 sampel pengujian uang dengan *database* citra latihan yang diperoleh baik dengan metode PCA maupun LDA. Citra pengujian terdiri dari 4 nominal uang rupiah emisi 2016 dengan masing-masing 4 orientasi uang. Masing-masing orientasi uang untuk setiap nominal memiliki 5 sampel citra pengujian.

Penyusunan *database* untuk masing-masing metode menggunakan citra-citra uang yang sama. Citra latihan terdiri dari 7 sampel gambar untuk masing-masing orientasi setiap nominal uang yang digunakan. Total citra latihan adalah sebanyak 112 gambar. Setiap data citra latihan disusun berurut di dalam matriks seperti yang sudah dijelaskan sebelumnya. Citra latihan dideskripsikan pada tabel 4.7.

Tabel 4.7. Deskripsi citra latihan

No	Nominal	Orientasi	Kolom	Contoh gambar
1	Rp 10.000,00	1	1-7	
2		2	8-14	
3		3	15-21	
4		4	22-28	
5	Rp 20.000,00	1	29-35	
6		2	36-42	
7		3	43-49	
8		4	50-56	
9	Rp 50.000,00	1	57-63	

10		2	64-70		
11		3	71-77		
12		4	78-84		
13	Rp 100.000,00	1	85-91		
14		2	92-98		
15		3	99-105		
16		4	106-112		

Dari hasil latihan dengan metode PCA dan LDA, diperoleh hasil *mean* yang serupa untuk kedua metode, karena keduanya menggunakan perhitungan yang sama. Bentuk *mean* dari citra latihan dalam bentuk gambar ditunjukkan pada gambar 4.5.



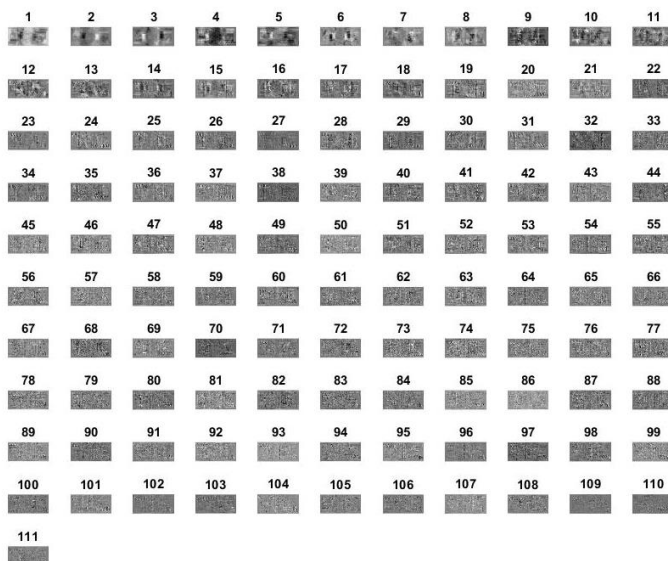
Gambar 4.5 *Mean* dari citra latihan dengan metode PCA dan LDA

Contoh citra uang yang dikurangi dengan *mean* dari citra-citra latihan ditunjukkan pada gambar 4.6.

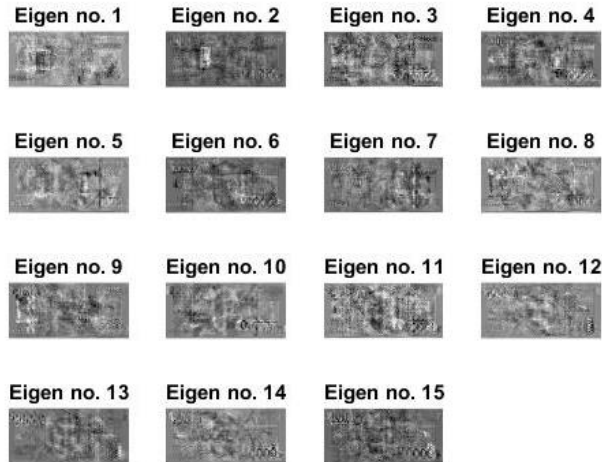


Gambar 4.6 (a) Citra awal. (b) Citra setelah dikurangi *mean*

Hasil pencarian *eigenvectors* dari PCA yang berjumlah 111 *eigenvectors* ditunjukkan pada gambar 4.7. *Eigenvectors* ini akan disusun untuk menjadi matriks transformasi PCA. Sedangkan hasil pencarian Fischer *eigenvectors* hasil metode LDA dengan jumlah 15 *eigenvectors* ditunjukkan pada gambar 4.8. Fischer *eigenvectors* ini akan disusun menjadi matriks transformasi LDA.



Gambar 4.7 *Eigenvectors* dari *database* metode PCA



Gambar 4.8 *Eigenvectors* dari *database* metode LDA

Dari kedua *database* citra latihan, dilakukan pengujian dengan matriks citra gabungan yang terdiri dari 80 citra pengujian. Citra pengujian terdiri dari 5 sampel untuk masing-masing 4 orientasi setiap nominal uang. Matriks citra pengujian gabungan ini disusun berurut sama seperti matriks citra latihan. Pengujian ini dilakukan untuk melihat akurasi pengenalan nominal uang sekaligus untuk membandingkan performa dari masing-masing metode. Hasil pengujian juga akan menentukan metode yang akan digunakan dalam sistem pembayaran.

Untuk meringkas hasil pengujian, maka diambil rata-rata jarak Euclidean untuk setiap orientasi sampel untuk masing-masing nominal uang. Dari hasil gagal deteksi tiap pengujian dapat diperoleh akurasi untuk masing-masing orientasi tiap nominal sampel. Hasil seluruh pengujian kemudian disimpulkan dengan mengambil rata-rata seluruh jarak Euclidean dari sampel pengujian, menghitung akurasi total pengujian, total gagal, dan standar deviasi sampel dari total pengujian. Digunakan standar deviasi sampel karena pengujian berperan sebagai sampel dari populasi pengujian yang tidak terbatas. Standar deviasi sampel menunjukkan keberagaman jarak Euclidean yang diperoleh. Hasil pengujian untuk masing-masing metode dijelaskan pada tabel 4.8 sampai tabel 4.11.

Tabel 4.8. Pengujian citra uang dengan *database* PCA

No	Nominal	Or	Jumlah citra	Rata-rata jarak(d_{\min})	Error	Akurasi
1	Rp 10.000,00	1	5	1302,63	0	100%
2		2	5	1356,56	0	100%
3		3	5	1164,53	0	100%
4		4	5	1508,21	0	100%
5	Rp 20.000,00	1	5	1174,69	0	100%
6		2	5	1102,9	0	100%
7		3	5	1462,56	1	80%
8		4	5	1230,81	0	100%
9	Rp 50.000,00	1	5	1108,33	0	100%
10		2	5	1242,48	1	80%
11		3	5	1315,84	0	100%
12		4	5	1325,81	0	100%
13	Rp 100.000,00	1	5	1266,26	0	100%
14		2	5	1371	0	100%
15		3	5	1226,28	0	100%
16		4	5	1143,11	0	100%

Tabel 4.9. Kesimpulan pengujian citra uang dengan *database* PCA

Total pengujian	Rata-rata jarak(d_{\min}) total	Standar deviasi sampel	<i>Distance Threshold</i>	Total error	Akurasi
80	1268,88	311,9	1580,78	2	97,5%

Dari data pengujian yang diperoleh, dapat dilihat bahwa terdapat 2 *error* dalam proses pengenalan nominal citra uang. Dari hasil keseluruhan kalkulasi jarak Euclidean, diperoleh rata-rata jarak dari 80 sampel adalah 1268,88 dengan standar deviasi sampel 311,9. Hasil rata-rata seluruh jarak Euclidean terkecil dan standar deviasi sampel dapat dijumlahkan untuk menentukan *distance threshold*. Untuk pengujian ini, diperoleh *distance threshold* sebesar 1580,78. *Distance threshold* menandakan batas jarak Euclidean minimum yang terdeteksi sebagai nominal uang yang terdapat pada *database*. Akurasi total dari pengujian dengan metode PCA adalah 97,5%.

Tabel 4.10. Pengujian citra uang dengan *database* LDA

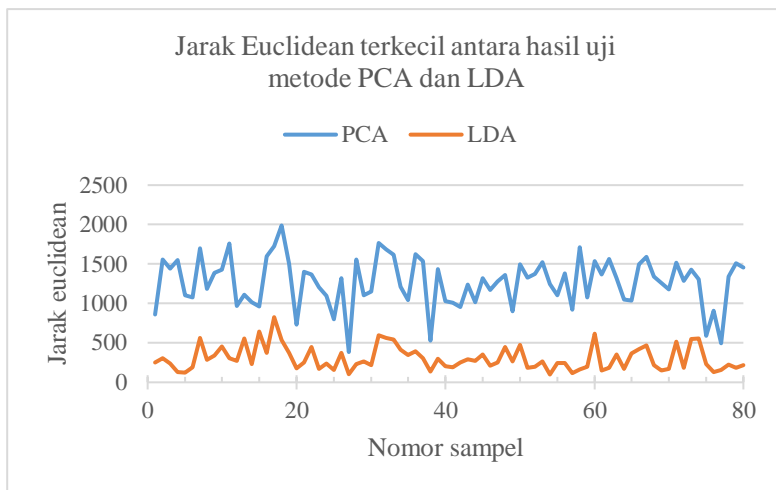
No	Nominal	Or	Jumlah citra	Rata-rata jarak(d_{\min})	Error	Akurasi
1	Rp 10.000,00	1	5	208,54	0	100%
2		2	5	366,98	0	100%
3		3	5	402,25	0	100%
4		4	5	457,06	0	100%
5	Rp 20.000,00	1	5	252,41	0	100%
6		2	5	236,47	0	100%
7		3	5	490,66	0	100%
8		4	5	267,57	0	100%
9	Rp 50.000,00	1	5	271,87	0	100%
10		2	5	328,91	0	100%
11		3	5	196,61	0	100%
12		4	5	267,25	0	100%
13	Rp 100.000,00	1	5	244,28	0	100%
14		2	5	285,63	0	100%
15		3	5	407,04	0	100%
16		4	5	183,81	0	100%

Tabel 4.11. Kesimpulan pengujian citra uang dengan *database* LDA

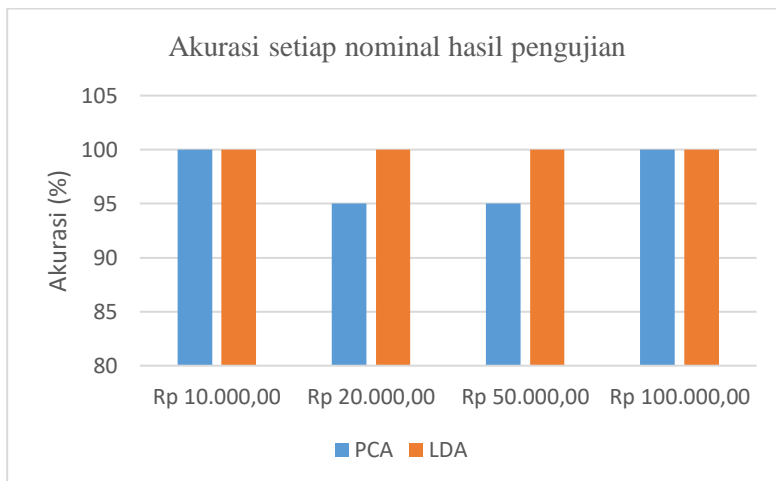
Total pengujian	Rata-rata jarak(d_{\min}) total	Standar deviasi sampel	<i>Distance Threshold</i>	Total error	Akurasi
80	304,21	152,6	456,81	0	100%

Dari data pengujian pada LDA, diperoleh akurasi pengenalan sebesar 100% karena tidak terdapat *error* dalam pengujian. Jarak Euclidean yang diperoleh juga jauh lebih kecil dibandingkan jarak Euclidean pada PCA. Standar deviasi sampel yang diperoleh juga lebih kecil, menandakan bahwa variasi jarak Euclidean berdasarkan sampel lebih kecil dibandingkan metode PCA. Dari kalkulasi jarak Euclidean rata-rata dan standar deviasi sampel, diperoleh *distance threshold* sebesar 456,81.

Perbandingan data pengujian PCA dan LDA ditunjukkan pada grafik pada gambar 4.9 dan 4.10.



Gambar 4.9 Perbandingan jarak Euclidean hasil uji PCA dan LDA



Gambar 4.10 Perbandingan akurasi hasil uji PCA dan LDA










Dari hasil perbandingan data hasil uji, maka dapat disimpulkan bahwa metode LDA lebih baik untuk mengenali nominal citra uang. Maka dari itu, metode analisa citra digital yang digunakan untuk mengenali








nominal citra uang pada sistem pembayaran mesin penjual swalayan pada tugas akhir ini adalah metode LDA. Selain itu, *distance threshold* yang digunakan juga berdasarkan hasil pengujian LDA.

4.8. Pengujian Dengan Uang Palsu

Setelah diputuskan metode sistem yang digunakan yakni metode LDA, maka juga dilakukan pengujian dengan uang palsu. Pengujian dilakukan dengan menghitung jarak Euclidean untuk masing-masing orientasi setiap nominal uang yang digunakan. Hasil deteksi dan jarak Euclidean minimum yang terdeteksi ditunjukkan pada tabel 4.12.

Tabel 4.12. Pengujian LDA dengan citra uang palsu

No	Nominal	Or	Gambar	Jarak terkecil(d_{min})	Hasil deteksi
1	Rp 10.000,00	1		1556,9	ditolak
2		2		1292,6	ditolak
3		3		1597,1	ditolak
4		4		1405,9	ditolak
5	Rp 20.000,00	1		1176,7	ditolak
6		2		1080,3	ditolak
7		3		1323,8	ditolak
8		4		1259,1	ditolak
9	Rp 50.000,00	1		1200,3	ditolak

10		2		1349,7	ditolak
11		3		1105,7	ditolak
12		4		1292,8	ditolak
13		Rp 100.00,00	1		1390,5
14	2			1275,3	ditolak
15	3			1101,5	ditolak
16	4			1157,3	ditolak
Rata-rata jarak euclidean				1285,3	

Dari hasil analisa, dapat disimpulkan bahwa sistem mampu menolak uang palsu. Semua jarak *Euclidean* yang diperoleh untuk masing-masing pengujian lebih besar dari *distance threshold* yang ditentukan (456,81) atau dengan kata lain semua pengujian menghasilkan hasil yang diinginkan yaitu penolakan semua uang palsu.

4.9. Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem dilakukan dalam 2 tahap, yaitu pengujian rekognisi sampel-sampel citra uji setiap nominal uang dengan menggunakan mode rekognisi pada sistem dan pengujian sistem pembayaran mesin penjual swalayan secara keseluruhan. Pengujian keseluruhan sistem dilakukan untuk menguji hasil akhir dari sistem yang telah dirancang pada tugas akhir ini.

4.9.1. Pengujian Rekognisi Citra Uang Sistem

Untuk kepentingan pengujian, maka diimplementasikan mode rekognisi yang hanya menampilkan hasil rekognisi citra uang pada tampilan LCD. Pengujian ini dilakukan dengan memasukkan 15 sampel uang untuk setiap nominal uang dengan orientasi acak. Hasil pengujian

ini ditunjukkan pada tabel 4.13.

Tabel 4.13. Hasil pengujian rekognisi citra uang

No	Nominal	Jumlah sampel	Hasil			Akurasi
			Benar	Salah	Ditolak	
1	Rp 10.000,00	15	15	0	0	100%
2	Rp 20.000,00	15	14	0	1	93,3%
3	Rp 50.000,00	15	12	1	2	80%
4	Rp 100.000,00	15	14	0	1	93,3%
Total		60	55	1	4	91,67%

Dari data pengujian, diperoleh akurasi 100% pada nominal Rp 10.000,00 dengan total 15 sampel uang direkognisi dengan benar oleh sistem. Pengujian untuk nominal Rp 20.000,00 menghasilkan hasil rekognisi benar sebanyak 14 dan 1 sampel ditolak, dengan akurasi sebesar 93,3%. Ketika sampel yang ditolak diuji coba kembali, rekognisi menghasilkan hasil rekognisi yang benar. Pengujian untuk nominal Rp 50.000,00 menghasilkan hasil 12 rekognisi benar, 1 salah, dan 2 ditolak. Satu sampel direkognisi sebagai Rp 20.000,00, sedangkan pengujian kembali sampel yang ditolak menghasilkan hasil rekognisi yang berhasil untuk kedua sampel. Rekognisi nominal Rp 100.000,00 menghasilkan 14 sampel berhasil direkognisi dengan benar dan 1 sampel ditolak. Ketika dilakukan pengujian kembali untuk sampel yang ditolak, diperoleh hasil rekognisi yang benar.

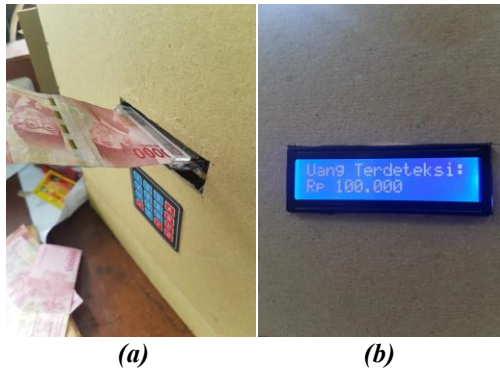
Hasil akhir dari pengujian ini adalah 55 sampel berhasil, 1 gagal, dan 4 sampel ditolak, dengan akurasi total 91,67%. Jika rekognisi yang ditolak dapat diabaikan dan dianggap sebagai rekognisi yang benar, maka akurasi yang diperoleh dari pengujian ditunjukkan pada tabel 4.14.

Tabel 4.14. Akurasi dengan pengujian ulang

Total sampel	Benar	Salah	Akurasi
60	59	1	98,3%



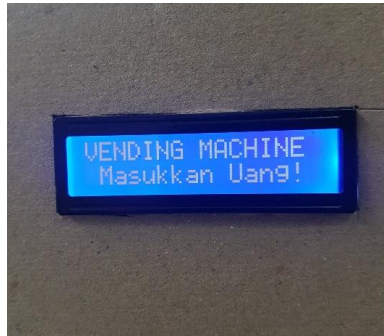
Gambar 4.11 Tampilan mode rekognisi pada LCD



Gambar 4.12 (a) Pengujian dengan uang Rp 100.000,00. (b) Hasil rekognisi pada LCD

4.9.2. Pengujian Sistem Pembayaran Mesin Penjual Swalayan

Pengujian sistem pembayaran mesin penjual swalayan secara keseluruhan dilakukan dengan melakukan transaksi pada sistem pembayaran secara kontinu dari tahap pemasukan uang sampai akhir transaksi. Tahapan yang diuji meliputi proses pemasukan uang, rekognisi citra uang, kalkulasi saldo, *input* perintah berupa transaksi, pembatalan transaksi, dan permintaan kembalian, dan percobaan beberapa kasus dalam sistem.



Gambar 4.13 Tampilan awal mesin penjual swalayan

Ketika pertama kali dinyalakan, LCD akan menampilkan tampilan awal seperti yang ditunjukkan pada gambar 4.13. Pada kondisi ini, *keypad* tidak akan menerima *input* apa pun kecuali pemilihan mode ke mode rekognisi. Ketika *keypad* ditekan sebelum uang dimasukkan, LCD akan menampilkan tampilan untuk memasukkan uang terlebih dahulu. Tampilan tersebut ditunjukkan pada gambar 4.14.

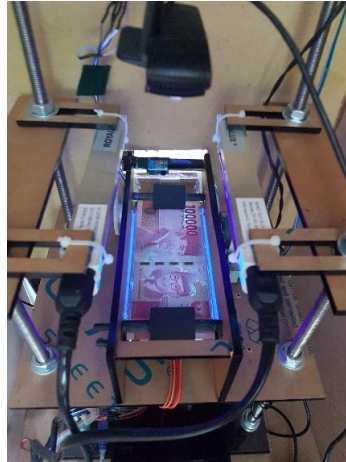


Gambar 4.14 Tampilan LCD ketika *keypad* ditekan sebelum ada uang yang masuk

Pengujian dimulai dengan memasukkan uang dengan nominal Rp 100.000,00 pada mulut *scanner* uang. *Scanner* secara otomatis akan mendeteksi keberadaan uang dan menerima uang dengan memutar motor pada *scanner* uang. Ketika uang sudah selesai dimasukkan, pengambilan citra uang akan dilakukan oleh kamera. Proses ini ditunjukkan pada gambar 4.15. Hasil dari akuisisi citra oleh kamera pada pengujian ini ditunjukkan pada gambar 4.16.



(a)



(b)

Gambar 4.15 (a) Proses pemberian uang Rp 100.000,00. (b) Posisi uang di dalam *scanner* uang.



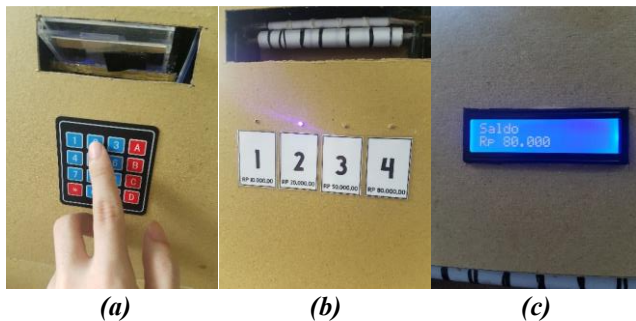
Gambar 4.16 Citra uang yang diakuisisi oleh kamera

Proses dilanjutkan dengan rekognisi citra uang pada MATLAB. Hasil rekognisi diteruskan ke Arduino Mega 2560 untuk dikalkulasi dengan saldo sebelumnya. Pada kondisi ini, saldo awal masih bernilai 0 sehingga saldo yang terdeteksi adalah Rp 100.000,00. Nominal saldo yang terdeteksi kemudian ditampilkan pada LCD.



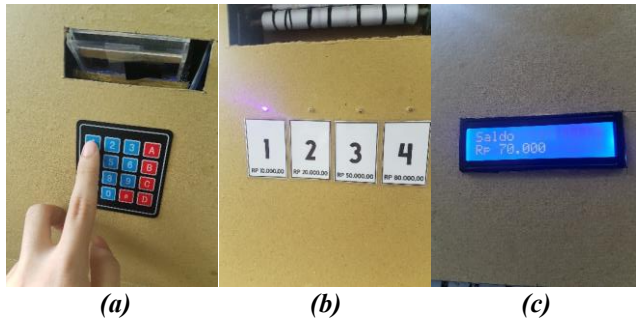
Gambar 4.17 Saldo hasil deteksi pengujian

Saldo yang sudah diterima dapat langsung digunakan untuk melakukan pembelian barang dari mesin penjual swalayan. Pada pengujian ini, dilakukan pembelian barang nomor 2 dengan harga Rp 20.000,00. Karena saldo yang tersedia mencukupi untuk melakukan transaksi, maka LED nomor 2 akan menyala menandakan transaksi dengan barang nomor 2 berhasil dan motor *scanner* akan memasukkan uang ke dalam penyimpanan uang di dalam bodi sistem.



Gambar 4.18 (a) Pemilihan harga barang 2. (b) LED tanda transaksi untuk barang 2 berhasil. (c) Sisa saldo setelah transaksi.

Setelah transaksi selesai, jumlah saldo yang tersisa akan dikalkulasi dan ditampilkan pada LCD. Saldo yang tersisa dapat digunakan kembali untuk membeli pilihan barang yang lain atau meminta kembalian uang dengan menekan tombol 'B'. Pada pengujian kali ini, akan dilakukan transaksi kembali dengan barang nomor 1 dengan harga Rp 10.000,00.



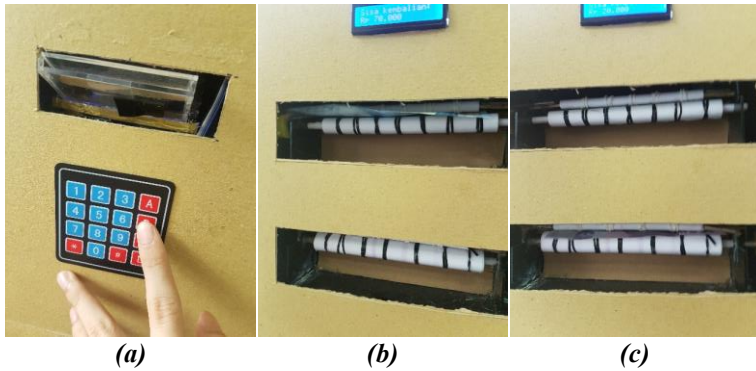
Gambar 4.19 (a) Pemilihan harga barang 1. (b) LED tanda transaksi untuk barang 1 berhasil. (c) Sisa saldo setelah transaksi.

Sama seperti sebelumnya, saldo masih dapat digunakan untuk melakukan transaksi. Untuk keperluan pengujian kasus, dilakukan permintaan pembelian barang nomor 4 dengan harga Rp 80.000,00. Karena saldo tidak mencukupi, maka sistem akan menolak transaksi dengan menampilkan indikasi bahwa saldo tidak cukup pada LCD.



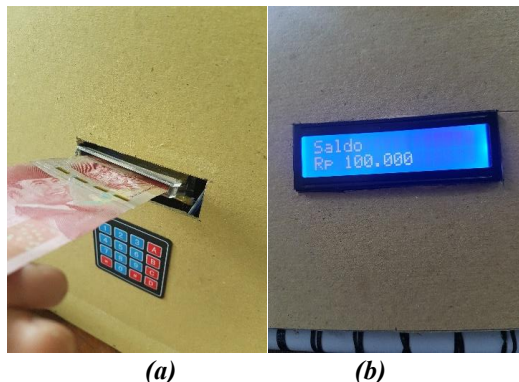
Gambar 4.20 Tampilan LCD ketika saldo tidak mencukupi

Ketika masih ada saldo yang tersisa, pengguna dapat meminta kembalian uang dengan menekan tombol 'B'. Pada pengujian kali ini, dilakukan permintaan kembalian uang dari hasil transaksi pada pengujian sebelumnya sebesar Rp 70.000,00. Pengembalian akan dilakukan satu-persatu dimulai dari pengembalian Rp 50.000,00 dan dilanjutkan dengan Rp 10.000,00 dengan jumlah uang sesuai dengan sisa saldo. Sisa kembalian ditampilkan pada LCD.



Gambar 4.21 (a) Permintaan kembalian. (b) Proses kembalian Rp 50.000,00. (c) Proses kembalian Rp 10.000,00.

Pada tahap ini, transaksi sudah selesai dan sistem kembali menunjukkan tampilan awal pada LCD. Untuk keperluan pengujian fitur penambahan dan pembatalan saldo, maka dilakukan pengujian dengan memasukkan uang Rp 100.000,00 terlebih dahulu, dan dilanjutkan pemasukan uang Rp 50.000,00. Saldo akan diakumulasi dengan uang yang terdeteksi sebelumnya, dan proses penambahan saldo ini dapat dilakukan kapan saja, baik sebelum maupun sesudah transaksi.



Gambar 4.22 (a) Pemasukan uang Rp 100.000,00. (b) Saldo awal terdeteksi.



(a) (b)

Gambar 4.23 (a) Penambahan uang Rp 50.000,00. (b) Saldo setelah penambahan uang.

Selain itu juga terdapat fitur pembatalan transaksi untuk membatalkan uang yang terakhir masuk ke dalam mesin dengan menekan *keypad* 'A'. Pembatalan dilakukan dengan mengurangi saldo dengan uang yang terakhir dideteksi dan memutar motor *scanner* uang ke arah sebaliknya. Fitur pembatalan hanya tersedia untuk uang terakhir yang dimasukkan dan transaksi pembelian barang belum dilakukan. Ketika tombol fitur pembayaran ditekan setelah transaksi, maka LCD akan menampilkan peringatan bahwa pembatalan sudah tidak dapat dilakukan.



(a) (b) (c)

Gambar 4.24 (a) Pembatalan transaksi. (b) Uang Rp 50.000,00 yang dikembalikan. (c) Pengurangan saldo ke kondisi sebelumnya

Dari hasil pengujian sistem secara keseluruhan, dapat disimpulkan bahwa sistem sudah dapat melakukan kinerjanya untuk melakukan transaksi tunai pada mesin penjual swalayan dengan baik. Seluruh fitur yang diprogram pada sistem juga sudah bekerja sesuai dengan yang diinginkan.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan percobaan yang telah dilakukan pada pelaksanaan tugas akhir ini didapat beberapa kesimpulan sebagai berikut:

- a. *Principal Component Analysis* (PCA) dan *Linear Discriminant Analysis* (LDA) yang umumnya digunakan dalam pengenalan citra wajah mampu bekerja dengan baik dalam mengenali nominal citra uang.
- b. Metode *Linear Discriminant Analysis* (LDA) memiliki akurasi pengenalan citra uang yang lebih baik dibandingkan dengan metode *Principal Component Analysis* (PCA).
- c. Nominal citra uang yang diterapkan masih terbatas untuk 4 nominal rupiah karena keterbatasan *database* yang disusun. Variasi nominal citra uang yang dapat dikenali dapat ditingkatkan dengan menambah jenis citra dalam *database*.
- d. Akurasi sistem rekognisi citra uang yang diterapkan pada sistem berdasarkan hasil pengujian sistem adalah 98,3% dari 60 sampel pengujian.
- e. Sebagian besar sistem yang dibangun sudah bekerja sesuai yang diinginkan, namun masih terdapat kelemahan pada mulut *scanner* uang dan mekanik pengembalian, di mana dapat terjadi tersangkutnya uang pada mekanik mulut *scanner* ketika pengembalian uang; dan mekanik pengembalian yang masih sangat sederhana dan rentan oleh terjadinya *error*.

5.2. Saran

Sebagai sarana pengembangan Sistem Pembayaran Tunai pada Mesin Penjual Swalayan ini, maka terdapat beberapa saran dari penulis berdasarkan hasil yang diperoleh saat percobaan, yaitu sebagai berikut:

- a. Pengujian dan analisa lebih lanjut dengan jumlah *database* latihan dan citra pengujian yang lebih banyak untuk meningkatkan akurasi pada kedua metode PCA dan LDA.
- b. Diperlukan suatu mekanik pengembalian uang dengan mekanisme yang lebih baik untuk mencegah terjadinya *error* yang dilengkapi dengan pendeteksi sisa uang kembalian yang tersisa.

- c. Penyesunan mekanik mulut *scanner* uang yang lebih baik agar kemungkinan uang tersangkut ketika uang disalurkan ke arah sebaliknya dapat diminimalisir.

DAFTAR PUSTAKA

- [1] W. Puyati and A. Walairacht, "Efficiency Improvement for Unconstrained Face Recognition by Weightening Probability Values of Modular PCA and Wavelet PCA," in *2008 10th International Conference on Advanced Communication Technology*, 2008, vol. 2, pp. 1449–1453.
- [2] "Buy Seaga Infinity INF5S Snack Vending Machine - Vending Machine Supplies For Sale." [Online]. Available: <https://www.candymachines.com/Seaga-Infinity-INF5S-Snack-Vending-Machine-P2641.aspx>. [Accessed: 01-Dec-2018].
- [3] Departemen Pengelolaan Uang, *Ciri Keaslian Uang Rupiah Tahun Emisi 2016*. Bank Indonesia, 2017.
- [4] "Undang-Undang Nomor 7 Tahun 2011 Tentang Mata Uang | Kitab Undang-Undang Hukum Pidana." .
- [5] "ISO 21348:2007(en), Space environment (natural and artificial) — Process for determining solar irradiances." [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso:21348:ed-1:v1:en>. [Accessed: 28-Nov-2018].
- [6] S. Herman, *Industrial Motor Control*. Delmar Cengage Learning, 2009.
- [7] B. Aranjó, P. K. Soori, and P. Talukder, "Stepper motor drives for robotic applications," in *2012 IEEE International Power Engineering and Optimization Conference Melaka, Malaysia*, 2012, pp. 361–366.
- [8] "4 Phase ULN2003 Stepper Motor Driver PCB Datasheet." .
- [9] "Reflective Optical Sensor with Transistor Output." Vishay Semiconductors.
- [10] "2 Channel 5V 10A Relay Module Datasheet." .
- [11] EngineersGarage, "LCD," 7-2010. [Online]. Available: <https://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>. [Accessed: 29-Nov-2018].
- [12] T. Gustavson, *Camera: A History of Photography from Daguerreotype to Digital*. .
- [13] "4x4 Matrix Membrane Keypad." Parallax.
- [14] D. H. Ballard and C. M. Brown, *Computer vision*. Englewood Cliffs, N.J: Prentice-Hall, 1982.
- [15] P. Chakravorty, "What Is a Signal? [Lecture Notes]," *IEEE Signal*

- Processing Magazine*, vol. 35, no. 5, pp. 175–177, Sep. 2018.
- [16] C. Solomon and T. Breckon, *Fundamentals of digital image processing: a practical approach with examples in Matlab*. Chichester, West Sussex ; Hoboken, NJ: Wiley-Blackwell, 2011.
 - [17] C. M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006.
 - [18] K. Pearson, “LIII. On lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, Nov. 1901.
 - [19] H. Hotelling, “Relations Between Two Sets of Variates,” *Biometrika*, vol. 28, no. 3/4, p. 321, Dec. 1936.
 - [20] P. Marasamy and S. Sumathi, “Automatic recognition and analysis of human faces and facial expression by LDA using wavelet transform,” in *2012 International Conference on Computer Communication and Informatics*, 2012, pp. 1–4.
 - [21] J. A. Rice, *Mathematical statistics and data analysis*, 3. ed., internat. ed. Belmont, Calif: Thomson/Brooks/Cole, 2007.
 - [22] H. Kusuma, “PCA Face Recognition,” Sep-2018.
 - [23] E. Kreyszig, H. Kreyszig, and E. J. Norminton, *Advanced engineering mathematics*. Hoboken, NJ: Wiley, 2011.
 - [24] H. Kusuma, “Problem Solving Matrik dan Vektor,” Sep-2018.
 - [25] “Linear Discriminant Analysis,” *Dr. Sebastian Raschka*, 03-Aug-2014. [Online]. Available: https://sebastianraschka.com/Articles/2014_python_lda.html. [Accessed: 02-Dec-2018].
 - [26] J. Cohen, Ed., *Applied multiple regression/correlation analysis for the behavioral sciences*, 3. ed., [Nachdr.]. Mahwah, NJ: Erlbaum, 2010.
 - [27] “MATLAB,” *Wikipedia*. 01-Dec-2018.
 - [28] X. Zhang and X. Ren, “Two Dimensional Principal Component Analysis based Independent Component Analysis for face recognition,” in *2011 International Conference on Multimedia Technology*, 2011, pp. 934–936.
 - [29] Y. Pei, “Linear Principal Component Discriminant Analysis,” in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 2108–2113.
 - [30] “Arduino mega 2560 R3,” *ELECTROBIST*. .

LAMPIRAN A

Program Arduino Sistem Pembayaran Tunai Mesin Penjual Swalayan

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
#include <Keypad.h>

//inisialisasi variabel serial dan mode
int recValue;
int mode = 1;

// driver scanner
int enB = 12;
int in3 = 51;
int in4 = 49;

boolean arah = LOW;
boolean enable = 1;

//sensor depan
const int pinIRd = 53;
const int pinIRa = A0;
int IRvalueA = 0;
int IRvalueD = 0;

//sensor belakang
const int pinIRd_2 = 47;
const int pinIRa_2 = A1;
int IRvalueA_2 = 0;
int IRvalueD_2 = 0;

//Kembalian Atas
int IN1=50;
int IN2=48;
int IN3=46;
int IN4=44;
int pinOut = 52;
```

```

//Kembalian Bawah
int IN1_2=30;
int IN2_2=28;
int IN3_2=26;
int IN4_2=24;
int pinOut_2 = 22;

int delaytime=2;

//LED Output
int pinLED_1 = 42;
int pinLED_2 = 40;
int pinLED_3 = 38;
int pinLED_4 = 36;

// Inisialisasi variabel dan state
int state = 0;
int uang = 0;
int saldo = 0;
int transaksi = 0;

//keypad
const byte numRows= 4;
const byte numCols= 4;
char keymap[numRows][numCols]= { {'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'} };
byte rowPins[numRows] = {9,8,7,6}; //Rows 0 to 3
byte colPins[numCols]= {5,4,3,2}; //Columns 0 to 3
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins,
numRows, numCols);

void setup() {
  Serial.begin(9600);

  lcd.begin(16,2);
  lcd.backlight();

```



```

// pin sensor
pinMode(pinIRd,INPUT);
pinMode(pinIRa,INPUT);

// pin motor scanner
pinMode(enB, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);

// pin kembalian
pinMode(IN1,OUTPUT);
pinMode(IN2,OUTPUT);
pinMode(IN3,OUTPUT);
pinMode(IN4,OUTPUT);
pinMode(pinOut, OUTPUT);
pinMode(IN1_2,OUTPUT);
pinMode(IN2_2,OUTPUT);
pinMode(IN3_2,OUTPUT);
pinMode(IN4_2,OUTPUT);
pinMode(pinOut_2, OUTPUT);

digitalWrite(pinOut, HIGH);
digitalWrite(pinOut_2, HIGH);

// pin LED
pinMode(pinLED_1, OUTPUT); // tengah kiri
pinMode(pinLED_2, OUTPUT); // ujung kiri
pinMode(pinLED_3, OUTPUT); // tengah kanan
pinMode(pinLED_4, OUTPUT); // ujung kanan

digitalWrite(pinLED_1, LOW);
digitalWrite(pinLED_2, LOW);
digitalWrite(pinLED_3, LOW);
digitalWrite(pinLED_4, LOW);

showWelcome();

}

```

```

void loop() {

    char keypressed = myKeypad.getKey();

    IRvalueA = analogRead(pinIRa);
    IRvalueD = digitalRead(pinIRd);

    IRvalueA_2 = analogRead(pinIRa_2);
    IRvalueD_2 = digitalRead(pinIRd_2);

    recValue = Serial.read();

    if (keypressed != NO_KEY) {
        if (keypressed == '*') {
            mode = 0;
            showModeRec();
        }
        if (keypressed == 'A' && mode == 0) {
            motorkeluar(75);
            delay(1000);
            showModeRec();
        }
        if (keypressed == '#') {
            mode = 1;
            saldo = 0;
            uang = 0;
            showWelcome();
        }

        else if (saldo == 0 && mode == 1) {
            showNomoney();
            delay(2000);
            showWelcome();
        }
        else {
            if (keypressed == '1' && mode == 1) {
                if (saldo < 10) {
                    showKurang();
                }
            }
        }
    }
}

```

```

        delay(2000);
        showSaldo(saldo);
    }
    else {
        saldo = saldo - 10;
        showThx();
        transaksi = 1;
        digitalWrite(pinLED_2, HIGH);
        if (IRvalueD_2 == 0) {
            motormasuk(75);
            delay(2000);
            showSaldo(saldo);
            digitalWrite(pinLED_2, LOW);
        }
        else {
            delay(2000);
            showSaldo(saldo);
            digitalWrite(pinLED_2, LOW);
        }
    }
}
}
if (keypressed == '2' && mode == 1) {
    if (saldo < 20) {
        showKurang();
        delay(2000);
        showSaldo(saldo);
    }
    else {
        saldo = saldo - 20;
        showThx();
        transaksi = 1;
        digitalWrite(pinLED_1, HIGH);
        if (IRvalueD_2 == 0) {
            motormasuk(75);
            delay(2000);
            showSaldo(saldo);
            digitalWrite(pinLED_1, LOW);
        }
    }
}

```

```

        else {
            delay(2000);
            showSaldo(saldo);
            digitalWrite(pinLED_1, LOW);
        }

    }
}

if (keypressed == '3' && mode == 1) {
    if (saldo < 50) {
        showKurang();
        delay(2000);
        showSaldo(saldo);
    }
    else {
        saldo = saldo - 50;
        showThx();
        transaksi = 1;
        digitalWrite(pinLED_3, HIGH);
        if (IRvalueD_2 == 0) {
            motormasuk(75);
            delay(2000);
            showSaldo(saldo);
            digitalWrite(pinLED_3, LOW);
        }
        else {
            delay(2000);
            showSaldo(saldo);
            digitalWrite(pinLED_3, LOW);
        }
    }
}

if (keypressed == '4' && mode == 1) {
    if (saldo < 80) {
        showKurang();
        delay(2000);
        showSaldo(saldo);
    }
}

```

```

else {
    saldo = saldo - 80;
    showThx();
    transaksi = 1;
    digitalWrite(pinLED_4, HIGH);
    if (IRvalueD_2 == 0) {
        motormasuk(75);
        delay(2000);
        showSaldo(saldo);
        digitalWrite(pinLED_4, LOW);
    }
    else {
        delay(2000);
        showSaldo(saldo);
        digitalWrite(pinLED_4, LOW);
    }
}

}

if (keypressed == 'A' && transaksi == 0 && mode == 1) {
    saldo = saldo - uang;
    motorkeluar(75);
    showBatal();
    transaksi = 1;
    delay(2000);
    showSaldo(saldo);
}
else if (keypressed == 'A' && transaksi == 1 && mode == 1) {
    showGagal();
    delay(2000);
    showSaldo(saldo);
}
if (keypressed == 'B' && transaksi == 0 && mode == 1) {
    showBelumTransaksi();
    delay(2000);
    showSaldo(saldo);
}
else if (keypressed == 'B' && transaksi == 1 && mode == 1) {
    if (saldo >= 50) {

```

```

        while (saldo >= 50) {
            showKembalian(saldo);
            digitalWrite(pinOut, LOW);
            kembali_50();
            saldo = saldo - 50;
        }
        digitalWrite(pinOut, HIGH);
        showWelcome();
    }
    if (saldo >= 10 && saldo < 50) {
        while (saldo >= 10) {
            showKembalian(saldo);
            digitalWrite(pinOut_2, LOW);
            kembali_10();
            saldo = saldo - 10;
        }
        digitalWrite(pinOut_2, HIGH);
        showWelcome();
    }
    transaksi = 0;
}
}

if(recValue == 101) {
    uang = 10;
    transaksi = 0;
    saldo = saldo + uang;
    if (mode == 0) {
        showUang(uang);
    }
    else if (mode == 1) {
        showSaldo(saldo);
    }
}
if(recValue == 102) {
    uang = 20;
    transaksi = 0;
    saldo = saldo + uang;

```

```

        if (mode == 0) {
            showUang(uang);
        }
        else if (mode == 1) {
            showSaldo(saldo);
        }
    }
    if(recValue == 103) {
        uang = 50;
        transaksi = 0;
        saldo = saldo + uang;
        if (mode == 0) {
            showUang(uang);
        }
        else if (mode == 1) {
            showSaldo(saldo);
        }
    }
    if(recValue == 104) {
        uang = 100;
        transaksi = 0;
        saldo = saldo + uang;
        if (mode == 0) {
            showUang(uang);
        }
        else if (mode == 1) {
            showSaldo(saldo);
        }
    }
    if(recValue == 100) {
        showTolak();
        motorkeluar(75);
        delay(2000);
        if(saldo == 0) {
            if(mode == 1) {
                showWelcome();
            }
            else if (mode == 0) {
                showModeRec();
            }
        }
    }

```

```

    }
  }
  else if(mode == 1) {
    showSaldo(saldo);
  }
  else if(mode == 0) {
    showRecGagal();
  }
}

if (IRvalueD == 1 && state == 0){
  motormasuk(0);
}

else if (IRvalueD == 0) {
  motormasuk(75);
  state = 1;
  delayMicroseconds(50000);
}

if (IRvalueD == 1 && state == 1){
  motormasuk(0);

  if (IRvalueD_2 == 1) {

  }
  else if (IRvalueD_2 == 0) {
    Serial.println(1);
  }

  state = 0;
}

}

void showWelcome()

```



```

{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("VENDING MACHINE");
    lcd.setCursor(0, 1);
    lcd.print(" Masukkan Uang! ");
}

```

```

void showSaldo(int saldo)
{
    if (saldo < 100 && saldo > 0) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Saldo");
        lcd.setCursor(0, 1);
        lcd.print("Rp");
        lcd.setCursor(3,1);
        lcd.print(saldo);
        lcd.setCursor(5,1);
        lcd.print(".000");
    }
    else if (saldo >= 100) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Saldo");
        lcd.setCursor(0, 1);
        lcd.print("Rp");
        lcd.setCursor(3,1);
        lcd.print(saldo);
        lcd.setCursor(6,1);
        lcd.print(".000");
    }
    else if (saldo == 0) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("VENDING MACHINE");
        lcd.setCursor(0, 1);
        lcd.print(" Masukkan Uang! ");
    }
}

```

```

}

void showTolak()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Uang Ditolak");
}

void showKurang()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Saldo Kurang!");
}

void showThx()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Terima Kasih!");
}

void showBatal()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Transaksi Batal");
}

void showBelumTransaksi()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Transaksi belum");
    lcd.setCursor(0, 1);
    lcd.print("dilakukan!");
}

```

```

void showNomoney()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Masukkan uang");
    lcd.setCursor(0, 1);
    lcd.print("terlebih dahulu!");
}

void showGagal()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("GAGAL! Tidak ada");
    lcd.setCursor(0, 1);
    lcd.print("uang terdeteksi");
}

void showModeRec()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" MODE REKOGNISI ");
    lcd.setCursor(0, 1);
    lcd.print(" Masukkan uang! ");
}

void showRecGagal()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("    Uang tidak    ");
    lcd.setCursor(0, 1);
    lcd.print("    dikenal!    ");
}

void showUang(int saldo)
{

```

```

    if (saldo < 100 && saldo > 0) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Uang Terdeteksi:");
        lcd.setCursor(0, 1);
        lcd.print("Rp");
        lcd.setCursor(3,1);
        lcd.print(saldo);
        lcd.setCursor(5,1);
        lcd.print(".000");
    }
    else if (saldo >= 100) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Uang Terdeteksi:");
        lcd.setCursor(0, 1);
        lcd.print("Rp");
        lcd.setCursor(3,1);
        lcd.print(saldo);
        lcd.setCursor(6,1);
        lcd.print(".000");
    }
}

```

```

void showKembalian(int saldo)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sisa kembalian:");

    if (saldo < 100 && saldo > 0) {
        lcd.setCursor(0, 1);
        lcd.print("Rp");
        lcd.setCursor(3,1);
        lcd.print(saldo);
        lcd.setCursor(5,1);
        lcd.print(".000");
    }
    else if (saldo >= 100) {

```

```

        lcd.setCursor(0, 1);
        lcd.print("Rp");
        lcd.setCursor(3,1);
        lcd.print(saldo);
        lcd.setCursor(6,1);
        lcd.print(".000");
    }
}

void motormasuk(int speedval)
{
    digitalWrite(in3, arah);
    digitalWrite(in4, !arah);
    analogWrite(enB, speedval);
}

void motorkeluar(int speedval)
{
    digitalWrite(in3, !arah);
    digitalWrite(in4, arah);
    analogWrite(enB, speedval);
}

void stepperbawah(){

//step 4
step1_2();
delay(delaytime);

//step 3
step2_2();
delay(delaytime);

//step 2
step3_2();
delay(delaytime);

//step 1
step4_2();

```

```

delay(delaytime);
}

void stepperatas(){

//step 4
step4();
delay(delaytime);

//step 3
step3();
delay(delaytime);

//step 2
step2();
delay(delaytime);

//step 1
step1();
delay(delaytime);
}

void step1(){
digitalWrite(IN1,LOW);
digitalWrite(IN2,LOW);
digitalWrite(IN3,HIGH);
digitalWrite(IN4,HIGH);
}

void step2(){
digitalWrite(IN1,HIGH);
digitalWrite(IN2,LOW);
digitalWrite(IN3,LOW);
digitalWrite(IN4,HIGH);
}

void step3(){
digitalWrite(IN1,HIGH);
digitalWrite(IN2,HIGH);

```

```
digitalWrite(IN3,LOW);  
digitalWrite(IN4,LOW);  
}
```

```
void step4(){  
digitalWrite(IN1,LOW);  
digitalWrite(IN2,HIGH);  
digitalWrite(IN3,HIGH);  
digitalWrite(IN4,LOW);  
}
```

```
void step1_2(){  
digitalWrite(IN1_2,LOW);  
digitalWrite(IN2_2,LOW);  
digitalWrite(IN3_2,HIGH);  
digitalWrite(IN4_2,HIGH);  
}
```

```
void step2_2(){  
digitalWrite(IN1_2,HIGH);  
digitalWrite(IN2_2,LOW);  
digitalWrite(IN3_2,LOW);  
digitalWrite(IN4_2,HIGH);  
}
```

```
void step3_2(){  
digitalWrite(IN1_2,HIGH);  
digitalWrite(IN2_2,HIGH);  
digitalWrite(IN3_2,LOW);  
digitalWrite(IN4_2,LOW);  
}
```

```
void step4_2(){  
digitalWrite(IN1_2,LOW);  
digitalWrite(IN2_2,HIGH);  
digitalWrite(IN3_2,HIGH);  
digitalWrite(IN4_2,LOW);  
}
```

```
void kembali_10() {  
  for (int i = 0; i<512 ; i++) {  
    stepperbawah();  
  }  
}
```

```
void kembali_50() {  
  for (int i = 0; i<512 ; i++) {  
    stepperatas();  
  }  
}
```


LAMPIRAN B

Program MATLAB Recognisi Nominal Citra Uang

Program perform_lda_PhD.m

```
function modelLDA = perform_lda_PhD(X, ids, n);
%% Dummy
modelLDA=[];

%% Check inputs

%check number of inputs
if nargin <2
    disp('Wrong number of input parameters! The function requires at
least two input arguments.')
    return;
elseif nargin >3
    disp('Wrong number of input parameters! The function takes no
more than three input arguments.')
    return;
elseif nargin==2
    [a,b]=size(unique(ids));
    n = max([a,b])-1;
end

%check if ids is a vector
if isvector(ids)==0
    disp('The second parameter "ids" needs to be a vector.')
    return;
end

% check if n is not to big
[a,b]=size(unique(ids));
if n>max([a,b])-1;
    disp('The parameter "n" must not be larger than the number of
classes minus one. Decreasing "n"!')
    n = max([a,b])-1;
end

%check that each image in X has a class label
```

```

[a,b]=size(X);
if b~=length(ids)
    disp('The label vector "ids" needs to be the same size as the number
of samples in X.')
    return;
end

```

```

%we assume that the data is contained in the columns
disp(sprintf('The training data comprises %i samples (images) with %i
variables (pixels).', b, a))
disp('If this should be the other way around, please transpose the
training-data matrix.')

```

```

%% First we compute the PCA subspace

```

```

%first model output
modelLDA.P = mean(X,2); %this is the mean face
modelLDA.dim = n;

```

```

%center data
Fi = X-repmat(modelLDA.P,1,b); %if you have memory problems put
this in a for loop

```

```

%compute PCA transform using the Eigenface trick
[U,V,L] = svd(Fi*Fi);
clear V L %save some memory
PCA = normc(Fi*U(:,1:b-1)); %here are the eigenfaces - display a
column in image form to visualize them
clear U

```

```

%sphere the PCA vectors (so that Sww is whitened after projection)
tmp = sqrt(diag(PCA'*Fi*Fi*PCA)); %find whitening tranform
PCA = PCA./repmat(tmp',a,1);

```

```

%% We compute the LDA transform in the PCA subspace - Fisherfaces

```

```

%compute class conditional means

```

```

id_unique = unique(ids);
[c,d]=size(id_unique);
num_of_class = max([c,d]); %this is the number of classes

%compute class-conditional means
meanclass = zeros(a,num_of_class);
class_id = zeros(1,num_of_class);
for i=1:num_of_class
    [dummy,ind]=find(id_unique(i)==ids);
    meanclass(:,i) = mean(X(:,ind),2);
end

%compute Fiw
Fiw = zeros(a,b);
cont = 1;
for i=1:num_of_class
    [dummy,ind]=find(id_unique(i)==ids);
    num_of_sampl = length(ind);
    for j=1:num_of_sampl
        Fiw(:,cont) = X(:,ind(j))-meanclass(:,i);
        cont=cont+1;
    end
end

%compute Fib
Fib = meanclass-repmat(modelLDA.P,1,num_of_class);

%this would be in a loop for larger problems
% Fib =zeros(a,num_of_class);
% for i=1:num_of_class
%     Fib(:,i) = meanclass(:,i) - model.P;
% end

%compute dimensionality reduction needed for inversion
max_dim = b - num_of_class;
%if you have duplicated images or the images in your trainnig data is
% highly correlated this might not be true; in this case I would suggest
to
% check the actual rank, i.e., max_dim = rank(Fiw);

```

```

%compute scatter matrices and projection into PCA subspace
Sbb = PCA(:,1:max_dim)*Fib*Fib'*PCA(:,1:max_dim);
Sww = PCA(:,1:max_dim)*Fiw*Fiw'*PCA(:,1:max_dim);
clear Fib Fiw %save some memory

% %solve the eigenproblem - option 1 (fixed)
% [U1,D1,QP] = svd(inv(Sww)*Sbb);
% clear D1 QP

%solve the eigenproblem via whitening - option 2 (could be extended to
EFM I or II)
[U1,D1,QP] = svd(Sww); %eigneprblem
U1_scaled = normc(U1); %normalized to unit norm
clear U1_norm U1 D1 QP
%we could check now that U1_scaled produces a whitened matrix:
tmp = U1_scaled*Sww*U1_scaled;

%change space
Sbbb = U1_scaled*Sbb*U1_scaled;
[U1,D1,QP] = svd(Sbbb); %eigenproblem

%backtrack to produce Fisherfaces
fisher = PCA(:,1:max_dim)*U1_scaled*U1(:,1:num_of_class-1);
clear PCA U1 U1_scaled

%exclude all that correspond to non-zero eignevalues
modelLDA.W = normc(fisher(:,1:num_of_class-1));
modelLDA.train = modelLDA.W*Fi;

```

Program Main_LDA.m

```

clear; clc;

%Training database matrix acquisition

```

```

Database_Training = [];
for i=1:112
    s = sprintf('Training7/%i.i.jpg',i);
    X=(imread(s));
    X2=rgb2gray(X);
    X3=double(X2);
    X3=imresize(X3, [80,NaN]);
    [size_y,size_x] = size(X3);
    Database_Training = [Database_Training,X3(:)];
end

save('Database_Training.mat', 'Database_Training');

%Testing database matrix acquisition
Database_Testing = [];
for i=1:80
    m = sprintf('Testing7/%i.i.jpg',i);
    Q=(imread(m));
    Q2=rgb2gray(Q);
    Q3=double(Q2);
    Q3=imresize(Q3, [80,NaN]);
    [size_yy,size_xx] = size(Q3);
    Database_Testing = [Database_Testing, Q3(:)];
end
save('Database_Testing.mat', 'Database_Testing');

%PCA Model for Training Data
ids = [1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4 5 5 5 5 5 5 5
6 6 6 6 6 6 6 7 7 7 7 7 7 7 8 8 8 8 8 8 9 9 9 9 9 9 10 10 10 10 10 10
10 11 11 11 11 11 11 11 11 12 12 12 12 12 12 13 13 13 13 13 13 13 14
14 14 14 14 14 14 15 15 15 15 15 15 15 16 16 16 16 16 16 16];
modelLDA = perform_lda_PhD(Database_Training, ids, 15);

%Show mean
figure(1)
imshow(reshape(modelLDA.P,size_y,size_x),[]) % seharusnya ada 15
eigen
title('Mean')

```

```

%Show Eigen
figure(2)
for i=1:15
    subplot(4,4,i)
    imshow(reshape(modelLDA.W(:,i),size_y,size_x),[])
    title(sprintf('Eigen no. %i',i));
end

%Mean Reduction

T=[];
for i=1:80
    T(:,i) = Database_Testing(:,i) - modelLDA.P;
end

%Projection
Y = modelLDA.W'*T;

%Distance
distance = [];
for itest=1:80
    for itrain=1:112
        distance((itrain),(itest)) =
sqrt(sum((((modelLDA.train(:,(itrain))) - (Y(:,(itest))))).^2));
    end
end

%save('Database_training_mean_LDA.mat', 'T');
%save('Distance_2_LDA.mat', 'distance');

%csvwrite('distance_LDA_3.csv',distance)
%save('modelLDA.mat', 'modelLDA');

```

Program serialsensor.m

```

clc;
delete(instrfind);

```

```
s = serial('COM6','BaudRate',9600);
set(s,'Timeout',1);
fopen(s);
```

```
go = true;
while go
    bytes = get(s,'BytesAvailable');
    bytes = s.BytesAvailable;
    if (bytes >= 1)
        flushinput(s);
        prosesuang();
    end
    flushinput(s);
    pause(1);
end
```

Program detectuang.m

```
clear cam
%hImage = image(zeros(480,640,'uint8'));

cam = webcam(1);
cam.FocusMode = 'manual'
cam.Focus = 15;
cam.Exposure = -4;
cam.BacklightCompensation = 1;
cam.WhiteBalanceMode = 'manual'
cam.WhiteBalance = 4000;
cam.Gain = 255;

%rectangle('Position',[55 90 520 240],...
% 'EdgeColor', 'b',...
% 'LineWidth', 3,...
% 'LineStyle','-')

pause(1);
img = snapshot(cam);
%hold off;
img_crop = imcrop(img,[55 90 520 240]);
```

```
imwrite(img_crop,'D:\MATLAB_TA\PCA_jpg\Inprogress\1.jpg');
```

Program prosesuang.m

```
%Testing data matrix acquisition
```

```
Matrix_Testing = [];
```

```
%load('modelPCA.mat');
```

```
load('modelLDA.mat');
```

```
detectuang();
```

```
m = sprintf('Inprogress/1.jpg');
```

```
Q=(imread(m));
```

```
Q2=rgb2gray(Q);
```

```
Q3=double(Q2);
```

```
Q3=imresize(Q3, [80,NaN]);
```

```
[size_yy,size_xx] = size(Q3);
```

```
Matrix_Testing = [Matrix_Testing, Q3(:)];
```

```
%Mean Reduction
```

```
T=[];
```

```
T(:,1) = Matrix_Testing(:,1) - modelLDA.P;
```

```
%Projection
```

```
Y = modelLDA.W'*T;
```

```
distance = [];
```

```
for itrain=1:112
```

```
    distance((itrain),(1)) = sqrt(sum(((modelLDA.train(:,(itrain))) -  
(Y(:,(1))))).^2));
```

```
end
```

```
I = find(distance==min(distance(:)));
```

```
NM = min(distance);
```

```
if (NM > 700)
```

```
    fprintf(s, 100);
```

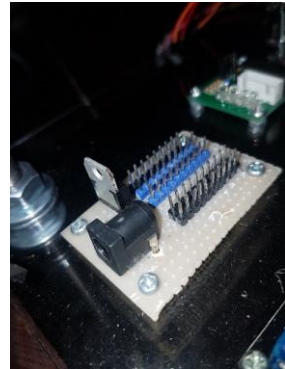
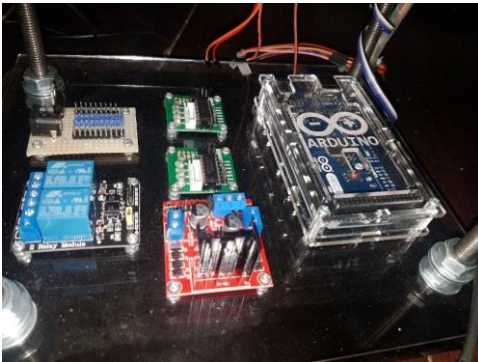


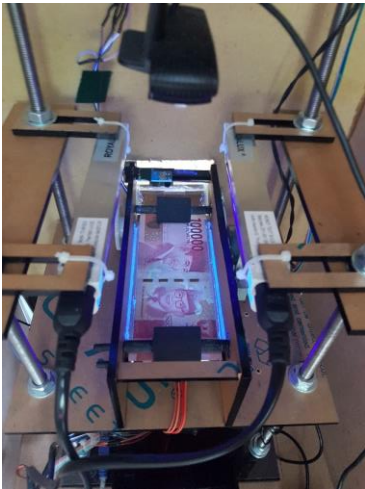
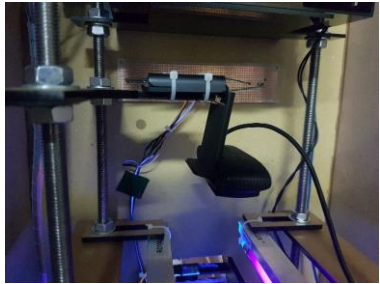
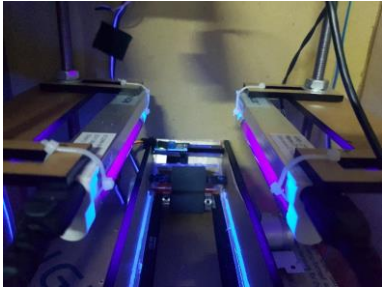
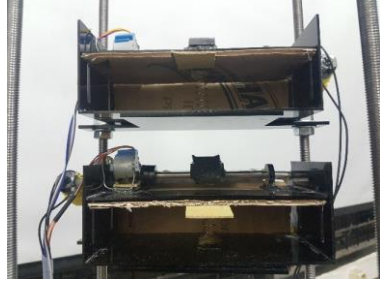
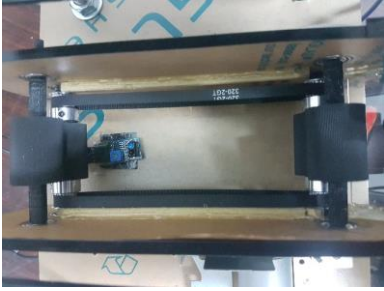
```
else
    if (I <= 28)
        hasil = 'uang 10.000';
        fprintf(s, 101);
    elseif (I >= 29) && (I <= 56)
        hasil = 'uang 20.000';
        fprintf(s, 102);
    elseif (I >= 57) && (I <= 84)
        hasil = 'uang 50.000';
        fprintf(s, 103);
    elseif (I >= 85) && (I <= 112)
        hasil = 'uang 100.000';
        fprintf(s, 104);
    end
end
```

.....*Halaman ini sengaja dikosongkan*.....

LAMPIRAN C

Dokumentasi Kegiatan Tugas Akhir







.....*Halaman ini sengaja dikosongkan*.....

BIODATA PENULIS



Fidelis Prasetyo lahir di Bekasi pada tanggal 24 April 1996 merupakan anak pertama dari empat bersaudara. Penulis menyelesaikan pendidikan dasar di SD Santo Vincentius Bidara Cina Jakarta, dilanjutkan dengan pendidikan tingkat menengah di SMP Santo Vincentius Bidara Cina Jakarta dan sekolah menengah atas di SMA Fons Vitae 1 Matraman Jakarta. Penulis memulai kehidupan perkuliahan pada tahun 2014 di Departemen Teknik Elektro, Fakultas

Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya. Selama masa perkuliahan, penulis aktif dalam berbagai kegiatan kepanitiaan dan organisasi mulai dari tingkat jurusan hingga nasional. Selain itu, penulis juga menjadi asisten praktikum di bidang studi elektronika.

Email : fidelisprasetyo@gmail.com
Hp/WA : 081231882446
Facebook : Fidelis Prasetyo
Line : fidelisprasetyo

.....*Halaman ini sengaja dikosongkan*.....